

망관리 객체의 시간지원 능동 특성에 대한 정형적 모델링

최 은 복[†] · 이 형 효^{**} · 노 봉 남^{***}

요 약

통신망 관리시스템은 다양한 구조와 특성을 가진 이질적인 통신망 구성요소를 효과적으로 감시, 제어하여 통신망을 효율적으로 운영하는 기능 외에, 사용자들로부터 요구되는 고도의 통신서비스를 신속하게 제공할 수 있어야 한다. 이를 위하여 ITU-T, ISO 등에 의해 제정된 표준권고안은 통신망 구성요소들간의 단순한 통신규칙을 정의하는 것 외에 통신망 관리에 필요한 자원들의 속성과 동적 특성에 대한 추상화된 표현, 그리고 통신망 구성요소들에 대한 관리기능을 포괄적으로 규정하고 있다. 그러나 표준통신망 구성요소를 기술하는 현재의 표준안이 관리객체의 구조나 속성 등 정적인 부분은 정형적으로 기술하는 데 반해 관리객체의 동적 특성에 대해서는 체계적으로 기술하지 못하고 있어, 관리객체의 전체적인 특성을 완전히 표현하지 못하는 문제점을 가지고 있다. 본 논문에서는 통신망 관리에 대한 표준권고안을 근거로 모든 통신망 관리객체에 공통적으로 적용될 수 있는 관리객체 동적 특성의 구성요소를 정의하고, 이들 구성요소간의 시간지원, 능동 특성을 기반으로 한 관리객체의 동적 특성에 대한 체계적이고 정형적인 기술방법을 제시한다.

A Formal Modeling for Temporal and Active Properties of Managed Object Behaviour

Eun-Bok Choi[†] · Hyung-Hyo Lee^{**} · Bong-nam Noh^{***}

ABSTRACT

Network management system(NMS) provides not only effective monitoring and controlling of network which consists of heterogeneous network elements but prompt response to users' need for high-level communication services. Recommendations of ITU-T and ISO stipulate the managerial abstraction of static and dynamic characteristics of network elements, management functions as well as management communication protocol. But the current description method does not provide the formal mechanism for the behavioral characteristics of managed objects in clear manner but in natural language form, the complete specification of managed objects is not fully described. In this paper, we describe determinants for the behaviour of managed objects applicable to every managed object, and present a language for specifying behavioral aspects of managed objects based on their temporal and active properties.

1. 서 론

새로운 통신기술과 다양한 컴퓨터 네트워크의 결합

* 본 논문은 1997년도 전남대학교 학술연구비에 의해 연구되었음.

† 준 회원 : 전남대학교 대학원 전산학과

** 정 회원 : 전남대학교 대학원 전산학과

*** 통신회원 : 전남대학교 전산학과 교수

논문접수 : 1999년 4월 26일, 심사완료 : 1999년 8월 3일

으로 대규모화된 통신망의 효과적인 관리와 사용자들의 고급 통신서비스 수요에 대한 신속한 제공을 위하여 표준화된 통신망 관리기술은 핵심적인 요소이다. 이를 위하여 ITU-T와 ISO 등의 표준화 기구에 의해 제정된 권고안들은 통신망 구성요소들간의 단순한 프로토콜을 정의하는 것 외에 통신망 관리에 필요한 자

원들의 속성(attribute)과 동적 특성(behaviour)에 대한 추상화된 표현방법, 그리고 망 구성요소들에 대한 관리기능을 포괄적으로 규정하고 있다. 따라서 이러한 표준 권고안들을 근거로 구현된 통신망 관리시스템은 이질적인 통신망을 포함하는 전체 통신망의 상호운용성(interoperability) 제공과 통합적인 망관리 기능의 수행을 가능하게 한다[1, 21].

통신망 관리시스템은 이질적 통신기술과 컴퓨터 네트워크로 구성된 통신망의 통합적인 관리기능의 수행을 위하여 관리대상 통신망 구성요소들의 특성 중 통신망 관리에 필요한 정적, 동적 정보만을 추출, 추상화한 관리객체(managed object)로 모델링한다[22]. 한편 관리대상 통신망 구성요소는 교환기에 장착된 통신포트, ATM 스위치 등과 같은 물리적인 요소와 현재 활성상태인 채널의 수, 라우팅 테이블, QoS 값 등 논리적인 요소들로 구성된다.

관리객체 클래스(managed object class)는 통신망 구성 자원들의 정적 속성 및 동적 특성, 연관된 관리객체 클래스간의 상속관계(inheritance) 등에 대한 정보를 객체지향 개념을 기반으로 표현한다. 따라서 통신망 관리시스템은 관리객체 클래스에 정의된 속성과 연산을 통해 통신망 구성요소 내부의 구현방식과는 무관하게 관리대상 객체의 동작상태를 감시하거나 필요에 따른 적절한 제어기능을 수행할 수 있다. ITU-T의 X.722 (Guidelines for the Definition of Managed Objects: GDMO) 권고안은 관리객체 클래스의 정의를 위한 템플릿(template)들에 대해 기술하고 있다. GDMO 표현법은 9가지 템플릿을 이용하여 관리객체의 정적 속성과 동적 특성 외에, 통신망 관리시스템에 관리객체 내부에서 관리상에 주요한 사항이 발생했음을 알리는 통지(notification)에 대해서 자세히 기술하고 있다. 그러나 현재의 GDMO 표현은 관리객체의 구조와 속성 등을 포함한 대부분의 정적 특성을 적절히 표현하고 있는데 반해, 동적 특성을 표현하는 방법으로는 텍스트 형태의 자연어를 이용하고 있어 관리객체의 모든 특성을 완전하게 기술하지 못하는 문제점을 가지고 있다.

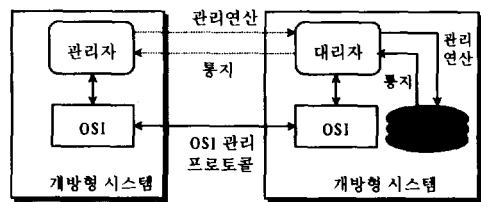
따라서 본 논문에서는 관리객체의 동적 특성을 기술하는데 사용되는 행위(BEHAVIOUR) 템플릿을 정형적으로 기술하기 위하여 표준권고안을 근거로 모든 관리객체에 공통적으로 적용될 수 있는 관리객체 동적 특성의 구성요소를 정의한다. 그리고, 이들 구성요소간의 시간지원, 능동 특성을 기반으로 하여 관리객체의 동

적 특성을 정확히 기술할 수 있는 정형적 기술방법을 제시한다.

본 논문의 구성은 다음과 같다. 2절에서는 관리정보 모델과 관리객체의 동적 특성에 대한 관련연구에 대해 살펴보고, 3절에서는 관리객체 동작특성의 구성요소에 대해 기술한다. 4절은 3절에서 정의된 관리객체 동적 특성의 정형적 기술방법을 포함하고 있으며, 5절의 결론에는 본 연구의 특징과 향후 연구과제를 기술하였다.

2. OSI 관리 프레임워크

통신망 관리시스템은 통신망을 구성하는 각 구성요소의 구성정보와 동작상태를 감시, 제어하고 향후 통신망의 설계에 활용될 수 있는 의사결정 정보를 제공하는데 있다. 이를 위하여 OSI 관리표준들은 개방형 시스템의 관리를 위한 시스템 관리의 기본원리 및 개념, 관리기능, 관리정보 체계에 대하여 기술하고 있다[1]. OSI 표준안들은 관리객체에 대한 표준화된 정의(ISO/IEC 10165-x), 시스템 관리기능에 대한 표준화된 정의(ISO/IEC 10164-x), 관리정보의 교환을 위한 표준화된 통신 서비스 및 프로토콜 사양(Common Management Information Service: CMIS/Common Management Information Protocol: CMP)에 대한 표준을 제정하여 통합망관리 시스템의 기능을 정의하고 있다[3, 4]. (그림 1)은 일반적인 개방형 시스템 관리체계를 나타내고 있다[5].



(그림 1) 시스템 관리 구조

2.1 관리정보 모델

관리객체는 통신망을 구성하는 다양한 구성요소들에 대하여 객체지향개념을 기반으로 설계되었으며 표준화된 관리 인터페이스 기능을 제공하고 있다[7,9]. 즉, 관리객체는 통신망 관리에 필요한 통신망 물리적 또는 논리적 구성요소를 추상화한 관리모델로서, 해당 망 구성요소의 특성을 표현하는 데이터와 그 망 구성요소에 적용될 수 있는 연산(operation)을 통신망 관리자에

게 제공하는 기능을 수행한다[23]. 그리고 유사한 데이터와 연산을 가지는 관리객체들의 그룹을 관리객체 클래스로 정의된다. ITU-T X.722 권고안에서는 객체지향 개념의 특성인 캡슐화, 정보은닉성, 상속성 등을 활용하여 관리객체 클래스를 정의함으로써 표준화된 통신망 관리시스템의 설계와 효과적인 개발을 지원하고 있다[7].

GDMO 표현법에서는 다음과 같은 9개의 템플릿(template)을 이용하여 관리객체 클래스의 정적 및 동적 특성을 기술하고 있다[8].

- MANAGED OBJECT CLASS
- PACKAGE
- ATTRIBUTE
- ATTRIBUTE GROUP
- ACTION
- PARAMETER
- NAME BINDING
- BEHAVIOUR
- NOTIFICATION

MANAGED OBJECT CLASS 템플릿은 관리객체 클래스를 정의하는 기능을 수행하며, 현재 정의되는 관리객체 클래스가 상속받는 하나 또는 그 이상의 상위클래스를 지정한다. 이 템플릿내에는 여러 개의 PACKAGE 템플릿을 포함할 수 있다. 상속성과 PACKAGE는 관리객체 클래스를 체계적으로 정의하고, 하나 이상의 관리객체 클래스에 적용될 수 있는 기능요소들을 재사용할 수 있도록 하는 메커니즘을 제공한다. PACKAGE 템플릿은 ATTRIBUTE, ATTRIBUTE GROUP, NOTIFICATION, ACTION, BEHAVIOUR, PARAMETER 템플릿에 의해 정의된 논리적으로 서로 연관된 구성요소들을 식별가능한 그룹으로 묶는 기능을 제공한다. PACKAGE 템플릿에는 'CHARACTERIZED BY'에 의해 지정되는 필수(mandatory) 패키지와 'CONDITIONAL PACKAGES'에 의해 지정되는 조건부(conditional) 패키지가 있으며 MANAGED OBJECT CLASS 템플릿에 포함될 수 있다. ATTRIBUTE 템플릿은 관리객체 클래스의 구성 속성들에 대한 구문(syntax)을 정의하고 있다. ATTRIBUTE 템플릿안에 포함되는 속성의 구문은 ASN.1[2] 자료형을 이용하여 표현된다. ATTRIBUTE GROUP 템플릿은 관리객체에 대한 편

리한 관리기능 수행을 위해 여러 개의 속성들을 하나의 단위로 취급하여 관리되게 하는 기능을 제공하고 있다.

ACTION 템플릿은 관리객체에 대해 GET 또는 REPLACE와 같이 미리 정의된 연산들에 의해 모델링되기 불가능한 연산을 정의하는 데 사용된다. PARAMETER 템플릿은 연산이나 통지 구문이나 CMIS 오류정보 등을 기술하는 데 사용되며, PACKAGE, ATTRIBUTE, ACTION, NOTIFICATION 템플릿에 포함될 수 있다. NAME BINDING 템플릿은 관리객체간 포함관계 및 상속 계층에서의 포함관계 적용범위, 포함관계에 있는 관리객체에 관리연산을 수행할 때의 제약사항 등을 기술한다.

NOTIFICATION 템플릿은 관리객체의 내부 또는 외부에서 특정 사건(event)이 발생하였음을 알리는 기능을 제공한다. 이러한 사건들은 관리객체에 따라 그 종류와 특성이 다르며, 지정된 사건이 발생하였음을 알리는 통지를 관리자에게 전송할 것인지에 대한 판단은 Event Forwarding Discriminator(EFD) 관리객체에 의해 결정된다.

관리객체의 동적 특성을 기술하는 데 사용되는 BEHAVIOUR 템플릿은 관리객체 자신(MANAGED OBJECT CLASS 템플릿)이나 관리객체를 구성하는 구성요소들(PACKAGE, PARAMETER, ATTRIBUTE, ACTION, NOTIFICATION, NAME BINDING 템플릿)에 대한 동적 특성을 기술하는 기능을 제공한다. 그러나 나머지 템플릿이 지정된 구문을 이용하여 관리객체의 특성을 기술하는 데 비해 BEHAVIOUR 템플릿은 자연어를 이용하여 구문이나 특정 형식이 없는 텍스트 형태로 기술하고 있어 관리객체의 동적 특성을 완전하고 정확하게 표현할 수 없는 문제점을 가지고 있다. 그 밖에도 관리객체의 동적 특성을 표현하기 위한 정형화된 형식이 정의되어 있지 않아서 BEHAVIOUR 템플릿을 이용한 프로그램의 자동생성 기능이나 동작기능의 시험 등의 자동화 작업이 불가능한 단점도 있다.

2.2 관련 연구

위에서 살펴본 관리객체의 동적 특성의 기술에 대한 문제점들을 해결하기 위하여 다양한 방식의 연구들이 진행되었다. Oliver는 관리객체의 동적 특성을 정형적으로 표현하기 위해 관리객체의 동작절차에 영향을 주는 요인들을 몇가지의 게이트(gate)로 정의하고, 관리

객체의 동작을 위한 조건(condition)과 수행 후의 변화(effects) 등을 규칙 형태로 정의한 표현방법을 제시하였다[16]. 또한 ER 모델을 기본으로 관련성이 있는 관리객체들간의 이벤트를 제약사항으로 표현한 ERC(Entity/Relationship/Constraints) 모델이 제시되었다[17]. 그러나 ERC 모델은 관리객체 기술의 표준인 GDMO와 다른 표현체계를 가지고 있어서 이미 GDMO 표현법에 의해 기술된 관리객체들과의 일관성이 결여된 문제점이 있다.

한편 Hasan은 통신망을 데이터베이스로 간주하고 통신망을 구성하는 관리객체에서 발생하는 이벤트 발생조건을 기술하는 ESL(Event Specification Language)을 제시하고, 발생한 이벤트에 대한 처리과정을 ECA(Event-Condition-Action) 규칙으로 모델링하였다[15]. 그러나 이 연구는 통신망 관리연산을 ECA 규칙으로 표현될 수 있음을 보이기 위해 관리객체로부터 생성된 이벤트들간의 관계와 그 처리방식을 중점으로 기술한 반면, 관리객체 내부에서 이벤트의 발생조건 및 처리절차 등 관리객체 수준의 동적 특성을 기술하지 못한 특징을 가지고 있다. Keller는 관리객체의 동적 특성을 정형적으로 기술하기 위해 GDMO 표현법을 확장하고, 그 실용성(feasibility)을 보이기 위해 새로운 템플릿을 정의하였다. 또한 새로 정의된 템플릿의 의미를 Z 명세언어를 이용하여 정형적으로 표현하였다[20]. 또한, 통신망 관리시스템의 정확한 기능 설계와 시험을 위하여 Object-Z와 RAISE 명세언어를 이용하여 관리객체를 정형적으로 표현하는 방법들과 각각의 특성을 비교한 연구가 진행되었다[14].

관리객체의 동적 특성을 파악하고 표현하기 위하여 관리객체 클래스간의 관계를 정의한 연구방법도 제시되었다. Clemm은 OSI 정보관리체계에 관리객체 클래스간 관계정보를 나타내는 관계 계층(relationship layer)을 추가하고, 관계를 추가하기 위한 템플릿을 제안하였다[18,19]. 한편 ITU-T X.725[10] GRM(General Relationship Model)은 관리객체 상호간의 관계(RELATIONSHIP CLASS)와 각 객체의 역할(ROLE BINDING)을 형식적으로 정의할 수 있는 규칙을 GDMO 표현법에 부가적으로 추가하여 기존의 OSI 정보모델에 영향을 주지 않으면서 관리객체간의 관계(relationship) 표현과 관련된 부분의 단점을 보완하고 있다. 그러나 이 방법 역시 RELATIONSHIP CLASS와 ROLE BINDING 템플릿의 동적 특성은 텍스트 형태의 자연어로 기술하

고 있다.

본 논문에서는 관리객체 자체의 동적 특성을 기술하는데 사용되는 행위(BEHAVIOUR) 템플릿을 정형적으로 기술하기 위하여 관리객체에 공통적으로 적용될 수 있는 동적 특성을 기술하기 위한 외부요인, 선행 및 불변조건, 그리고 동작절차를 결정하는 논리조건과 동기 발생에 대한 정형적 방법을 제시한다.

3. 관리객체의 동적 특성 모델링

통신망 관리 프로토콜을 통하여 전달되는 관리객체의 구조와 관리대상 자원의 관리적 특성들에 대해 기술하고 있는 ITU-T X.720 권고안에서는 관리객체 클래스 동적 특성이 다음과 같은 특성들을 포함하여 정의될 수 있다고 기술하고 있다[6].

- 속성, 연산, 동기 등의 의미(semantics)
- 관리객체에 대한 실행되는 연산에 대한 반응
- 관리연산의 정상적인 수행이나 동기가 발생되기 위한 선행조건(precondition) 및 후행조건(postcondition)
- 특정 속성값들간의 종속성 또는 속성에 대한 무결성 제약조건(integrity constraints)
- 관리객체간의 관련성(relationships)
- 관리객체의 생성부터 소멸 시점까지 항상 만족하여야 하는 불변조건(invariants)
- 관리객체의 동기(synchronization) 특성

그러나, 관리객체 클래스를 정의하기 위하여 사용되는 PACKAGE, ATTRIBUTE, ACTION, NOTIFICATION 등과 같은 템플릿들의 동작특성을 기술하는 BEHAVIOUR 템플릿이 위와 같은 특성을 표현할 수 있는 체계나 구문 등이 현재의 권고안에는 정의되어 있지 않아서, 관리객체의 동적 특성을 체계적이면서 명확하게 표현할 수 없는 문제가 있다. 그리고, 관리객체의 동적 특성 기술의 기준과 범위가 존재하지 않아, 관리객체마다 동적 특성을 기술하는 방법과 명세정도가 달라서 통신망 관리시스템 개발자에게 시스템 개발에 필요한 정보를 충분히 제공하지 못하였다. 따라서 본 논문에서는 ITU-T 권고안에서 정의하고 있는 관리객체 클래스의 동적 특성을 참조하여, 관리객체의 동적 특성에 영향을 미치는 구성요소를 정의하고 이를 체계적으로 표현할 수 있는 정형적 기술언어를 제시한

다. 또한, 제안된 동적 특성 기술 모델을 이벤트 보고 관리모델(event report management model)에 적용한 예를 보인다. 그림 2는 ITU-T X.734[12]에 정의된 이벤트 보고 관리기능을 기술하는 'eventForwardingDiscriminator' 관리객체 클래스와 그 상위클래스인 'discriminator' 관리객체 클래스에 대한 GDMO 표현을 나타내고 있다.

discriminator MANAGED OBJECT CLASS

DERIVED FROM top;

CHARACTERIZED BY

discriminatorPackage PACKAGE

BEHAVIOUR

discriminatorBehaviour BEHAVIOUR

DEFINED AS "This managed objects is used to represent the criteria for controlling management services.";

ATTRIBUTES

discriminatorID GET,

discriminatorConstruct

REPLACE-WITH-DEFAULT

DEFAULT-VALUE Attribute-ASN1Module.default Discriminator Construct **GET-REPLACE**,

CONDITIONAL PACKAGES

availabilityStatusPackage **PRESENT IF** "any of the scheduling packages, (duration, weekly scheduling, external) are present",

duration **PRESENT IF** "the discriminator function is scheduled to start at a specified time and stop at either a specified time or function continuously",

REGISTERED AS { smi2MObjectClass 3 };

eventForwardingDiscriminator MANAGED OBJECT CLASS

DERIVED FROM discriminator;

CHARACTERIZED BY

efdPackage PACKAGE

BEHAVIOUR

eventForwardingDiscriminatorBehaviour BEHAVIOUR

DEFINED AS "This managed object is used to represent the criteria that shall be satisfied by potential event reports before the event report is forwarded to a particular destination.";

ATTRIBUTES

destination **GET-REPLACE**:::

CONDITIONAL PACKAGES

backUpDestinationListPackage PACKAGE

ATTRIBUTES

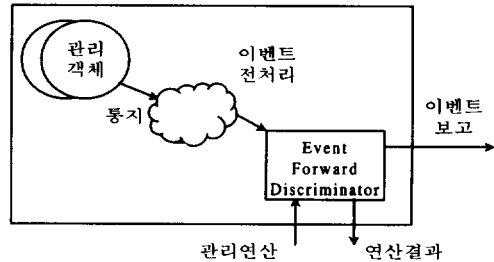
activeDestination **GET**;

backupDestinationList **GET-REPLACE**;

....

REGISTERED AS { smi2MObjectClass 4 };

(그림 2) discriminator, eventForwardingDiscriminator 관리객체 클래스의 GDMO 표현



(그림 3) 이벤트 보고 관리 모델

(그림 2)의 'eventForwardingDiscriminator' 관리객체 클래스는 'discriminator' 관리객체 클래스의 하위클래스로서, (그림 3)의 이벤트 전처리과정으로부터 생성된 이벤트 보고를 다른 관리객체(들)에게 이벤트 형태로 전달할 것인지를 결정하는 기능을 수행한다. 두 개의 관리객체 클래스 'discriminator'와 'eventForwardingDiscriminator'의 자세한 동적 특성이 X.734 권고안[12]에 상세히 기술되어 있으나, 그러한 특성을 정형적으로 표현할 수 있는 기술방법이 정해져 있지 않은 이유로 두 관리객체 클래스 모두 BEHAVIOUR 템플릿에서 자연어를 이용하여 관리객체의 동적 특성을 매우 포괄적인 내용으로 기술되어 있다.

3.1 동적 특성 기술모델의 요구사항

관리객체 동적 특성의 정형적 기술방법이 가져야 하는 특성 및 요구사항을 나열하면 다음과 같다.

첫째, ITU-T X.720[6] 표준 권고안에 정의된 관리객체 클래스의 동적 특성을 충실히 반영할 수 있는 기술구조를 지원한다.

둘째, 현재의 GDMO 표현체계를 준수하고 BEHAVIOUR 템플릿만을 최소한 수정하여 관리객체 클래스의 동적 특성을 기술하도록 한다.

셋째, 관리객체 클래스 설계자와 GDMO 표현법을 참조하여 통신망 관리기능을 구현하는 프로그램

래머가 쉽게 기술하고 이해할 수 있도록 프로그래밍 언어와 유사한 구조체를 사용한다.

넷째, 관리객체의 동적 특성을 기술하는 정형적 기술방법이 관리기능을 수행하는 프로그램의 자동생성이나 시험 등 과정에 활용될 수 있어야 한다.

그러나, 관리객체 클래스와 그 구성요소들이 가질 수 있는 모든 내포된 의미(semantics)와 동적 특성을 일반적(generic) 형태로 정형화하기는 매우 어려운 과정이다. 따라서 통신망의 운영과 관리에 관련되고 또한 모든 관리객체에 적용될 수 있는 관리객체의 주요동작 조건과 절차, 다른 관리객체들과의 관련성 등을 정형적인 방법으로 표현하는 작업은 관리객체의 동적 특성을 관리객체 클래스 설계자나 통신망 관리시스템 개발자들이 관리객체의 특성을 정확히 전달할 수 있는 장점을 가지고 있다.

3.2 관리객체 동적 특성의 모델링

관리객체 클래스에 대한 동적 특성을 모델링하는 과정은 OMT(Object Modeling Technique)의 동적 모델링(dynamic modeling) 과정과 유사하다. OMT의 객체모델(object model)이 클래스와 속성, 그리고 클래스간의 관계와 같은 정적인 정보를 표현하는데 비해 OMT 동적 모델은 객체모델에 의해 정의된 시스템이 시간과 외부 요인에 의해 그 상태가 변화되는 과정을 기술한다[13].

본 논문에서는 관리객체의 동작에 영향을 미치는 요인을 크게 외부 요인과 내부 요인으로 정의한다. 내부 요인중 주요한 요인으로는 관리객체를 구성하는 속성값이나 조건부 패키지의 포함여부, 그리고 별도로 정의된 연산(ACTION 템플릿)의 수행결과 등이 있다. 외부 요인으로는 통신망 관리자로부터 전달되는 관리연산(management operation) 실행요청과 다른 관리객체로부터 전달되는 통지 또는 이벤트 보고 등이 있다. 그리고 관리객체의 동작 결과에 따라 통지를 다른 시스템으로 생성하여 전송할 수 있다.

3.2.1 관리연산

관리객체에 대한 관리연산은 해당 관리객체 생성, 삭제 등과 같이 관리객체 자체를 직접 제어(CREATE, DELETE)하거나 관리객체의 동작에 영향을 미치는 속성값을 변화(REPLACE, REPLACE-WITH-DEFAULT,

ADD, REMOVE)시킴으로써 관리객체의 동적 특성을 변경시킨다. 예를 들면, ITU-T X.734[12]에 정의된 이벤트 보고 관리기능의 경우 이벤트에 대한 처리 기능을 제어하기 위해 discriminator 관리객체의 생성/삭제를 통해 이벤트 처리기능의 수행여부를 결정한다. 또한 discriminator 관리객체의 속성 중 'CMISFilter' 자료형을 갖는 'discriminatorConstruct' 속성값을 변경하여 이벤트 관리기능의 동작절차를 변경시킬 수 있다. 관리객체의 생성이나 삭제 연산이 정상적으로 수행된 경우 'objectCreation' 또는 'objectDeletion' 통지가 생성된 후, 관련된 관리객체들로 전송된다. 이때 전송되는 통지는 그 통지를 수신하는 관리객체의 동작절차를 결정하는 외부 요인으로서의 기능을 수행하게 된다(그림 2참조). 또한 관리연산의 수행과 이미 정의된 표준 관리연산 외에 관리객체마다 독특한 연산(ACTION) 수행중 발생하는 오류정보나 동작모드(MODE CONFIRMED) 역시 관리객체의 동적 특성 기술에 사용될 수 있다.

3.2.2 통지 또는 이벤트

관리객체에 의해 정의되고 발생되는 통지는 일반적으로 관리객체마다 다를 수 있으며, 관리객체로 전송되는 통지 또는 이벤트는 그 관리객체와 연관된 관리객체들로부터 생성되어 전달된다. 그러나 관리객체의 생성과 삭제, 동작상태 변화, 경보보고(alarm report), 보안 경보보고 등 대부분 관리객체에 적용되는 공통의 통지들이 시스템 관리기능 표준안에 정의되어 있다[7]. 관리객체로 전송된 통지나 이벤트, 그리고 함께 전달될 수 있는 매개변수값에 의해 관리객체의 동작특성이 영향을 받을 수 있다. 예를 들어, (그림 2)의 'eventForwardingDiscriminator' 관리객체의 경우 관리객체에 포함된 속성값 'discriminatorConstruct'에 의해 통지의 발생과 전달여부가 결정된다. 일반적으로 다른 관리객체에 의해 전달되는 통지는 관리객체의 특성에 따라 전처리과정(pre-processing)을 거쳐 처리 가능한 형태로 바뀌어 이벤트 보고 형식으로 연관된 관리객체에게 전송된다.

그러나, 통지의 발생여부를 결정하기 위한 이벤트 처리과정에 대한 명확한 기술방법이나 선택형 매개변수 생성에 대한 내용은 표준안에 기술되어 있지 않는 문제점이 있다[21].

3.2.3 관리객체의 속성 또는 조건부 패키지

관리객체의 동작특성은 위에서 살펴본 관리연산, 통지 또는 이벤트와 같은 외부적인 요인 외에 관리객체에 의해 직접 정의된 속성이나 다른 관리객체로부터 상속받은 속성, 그리고 필수 또는 조건부 패키지에 의해 결정된다. 관리객체에 전달된 외부요인에 따라 그에 대응되는 관리객체의 동작이 이루어지지만, 외부요인에 대한 처리과정은 관리객체의 동작상태를 나타내는 속성의 값이나 조건부 패키지의 유무에 따라 달라질 수 있다. 이처럼 관리객체의 동작과정은 ECA(Event-Condition-Action) 규칙에 따르는 능동적(active) 특성을 가진다.

관리객체의 동작상태를 표시하는데 이용되는 속성들의 예를 들면, 관리객체의 서비스 이용에 대한 허가여부를 나타내는데 사용되는 'administrativeState' 속성이 있다. 만일 'administrativeState' 속성값이 'locked'인 경우에는 'discriminator' 관리객체에 대한 어떠한 관리연산이나 동작도 허용되지 않는다. 관리객체의 동작상태를 나타내는 또 다른 속성 'operationalState'는 현재 관리객체의 동작상태를 제공하는 기능을 가지며, 그 속성값이 바뀌는 경우 동작상태가 바뀌었음을 알리는 통지가 관련된 관리객체들로 전송된다. 이벤트 보고 관리기능의 경우, 'administrativeState'와 'operational State' 속성 외에 스케줄링과 관련된 조건부 패키지의 포함 여부에 따라 관리객체의 동작이 달라지는 특징을 가진다.

3.2.4 시간자원 속성

관리객체의 동적 특성은 능동적 특성 외에 시간자원 특성에 따라 동작 절차가 달라질 수 있다. 관리객체의 동작은 통신망 관리 목적이거나 관리객체의 특성에 따라 주어진 기간 또는 반복적인 작동 주기를 가지고 동작될 수 있으며, 이를 정확히 표현할 수 있다면 관리객체의 능동적 특성을 보완하는 한편 매우 복잡한 운용 조건을 가지는 관리객체의 동적 특성을 정확히 기술할 수 있다. 예를 들어, 그림 2의 'discriminator' 관리객체 클래스의 경우 'discriminator' 관리객체에 전달된 이벤트는 'discriminator' 속성에 정의된 필터에 의해 이벤트 보고 생성 여부를 결정한다. 이때 이벤트 보고 발생을 결정하는 또 다른 요인이 스케줄링 패키지로서 시간제약 조건을 포함하고 있다. 스케줄링 패키지는 복잡한 이벤트 보고 주기(period)의 관리를 위해 정의

된 조건부 패키지로서, 'eventForwardingDiscriminator' 관리객체가 다양한 시간조건에 따라 동작할 수 있도록 제어하는 기능을 제공한다. 'availabilityStatusPackage' 패키지는 현재 관리객체의 가용상태 정보를 제공하며, 'duration' 패키지는 관리객체가 동작하는 시작 및 종료시각을 지정하여 관리객체 동작을 자동으로 제어할 수 있도록 한다. 'dailyScheduling' 패키지는 하루동안에 관리객체가 동작하는 시작과 종료시각을 여러 번 지정할 수 있는 기능을 제공하며, 'weeklyScheduling' 패키지는 일주일동안 관리객체가 동작하는 요일들과 그 요일에 대한 한 번 이상의 시작, 종료시각을 지정하는데 이용된다. 'externalScheduler' 패키지는 다른 패키지들과 같이 규칙적인 시간제약이 아닌 경우 외부 스케줄링 관리객체로부터 'discriminator' 관리객체의 동작과 중지에 대한 제어정보를 받게 된다.

앞에서 언급된 바와 같이, 지금까지 살펴본 관리객체에 동작에 영향을 미치는 능동적 또는 시간자원 요인들과 동작절차 및 그에 따른 통지 발생 등의 과정들이 BEHAVIOUR 템플릿내에 명확히 기술되어야 하나, 대부분 관리객체의 경우 동작 절차가 거의 기술되지 않거나 자연어를 이용하여 매우 포괄적이며 간략히 기술되어 관리객체의 동작 조건이나 절차, 결과 등이 명확히 표현되지 못한 문제점을 가지고 있다. 4절에서는 GDMO 표현법에서 현재의 BEHAVIOUR 템플릿 기술방법의 단점을 보완하기 위하여 이 절에서 살펴본 관리객체 동적 특성을 결정하는 구성요소들을 기반으로 한 정형적인 기술방법에 대해 설명한다.

4. 관리객체 동적 특성의 정형적 기술

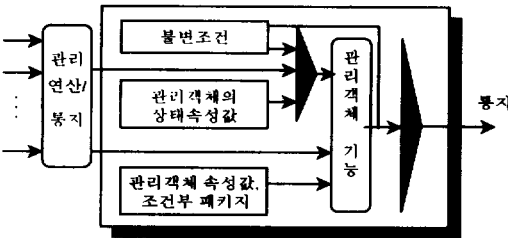
이 절에서는 ITU-T X.720에 정의된 관리객체 클래스의 동적 특성 요소들과 3.1에서 언급된 정형적 기술방법의 요구사항을 반영한 관리객체의 동적 특성에 대한 정형적 기술방법을 제시한다. 본 논문에서는 관리객체 동적 특성의 기술을 위한 주요 구성요소와 기술하기 위한 BDL(Behaviour Description Language for Managed Objects)을 정의한다(부록 참조). 프로그래밍 언어 형태의 정형적 기술방법은 관리객체의 동작 특성을 결정하는 요인들과 동작 과정을 명확히 기술함으로써 관리객체의 설계자나 프로그래머들이 관리객체의 동적 특성을 표현하고 이해하는데 도움이 되며, 관리객

체의 구현단계에서 관리객체의 주요 기능에 대한 프로그램의 자동생성에 활용될 수 있는 장점을 제공한다.

4.1 동작 절차의 결정요소 기술

본 논문에서는 관리객체 동적 특성의 기술을 위한 주요 결정요소로서 관리객체의 동작을 기동시키는 외부요인(EVENT), 정상적인 동작을 위한 선행조건(PRECOND), 관리객체가 항상 만족시켜야 하는 불변조건(INVARIANTS) 등으로 구성된다. 외부요인으로는 관리객체에 대한 관리연산과 다른 관리객체로부터 전달되는 통지 등이 있으며, 선행조건은 관리연산의 정상적인 수행이나 통지가 발생되기 위한 논리조건을 기술하게 된다. 불변조건은 관리객체의 동작 전체 과정에서 지켜져야 하는 조건을 기술한다. 선행조건과 불변조건의 기술에는 관리객체로 전달된 관리연산이나 통지에 포함된 매개변수와 관리객체에 정의된 속성값들 간의 종속성 또는 무결성 제약조건 등이 포함된다.

관리객체의 동작 과정은 위에서 설명한 주요 결정요소들과 제어흐름, 관리객체 속성의 변경이나 통지의 생성 및 전송 기능을 제공하는 연산자들을 이용하여 동작절차(PROCEDURE) 부분에서 기술된다.



(그림 4) 관리객체 동적 특성 결정요소

관리객체의 동작 절차의 결정요소의 외부요인인 관리연산과 이벤트 보고는 그 종류에 따른 매개변수가 함께 입력으로 전달될 수 있으며, 이벤트 보고 관리기능에서 'eventForwardingDiscriminator' 관리객체의 동작을 기동시키는 관리연산과 통지에 대한 기술을 예로 들면 다음과 같다.

eventForwardingDiscriminatorBehaviour BEHAVIOUR
DEFINED AS

```
EVENT: faultReportEvent
managedObjectClass: ObjectClass,
managedObjectInstance: ObjectInstance,
```

```
eventType: EventTypeID,
severity: SecurityAlarmSeverity,
backedUpStatus: BackedUpStatus,
probableCause: ProbableCause;
```

```
EVENT: replaceDiscriminator
inDiscriminatorConstruct: DiscriminatorConstruct;
```

(그림 5) 외부요인 기술의 예

(그림 5)의 'faultReportEvent'는 장애발생 보고에 관련된 외부요인으로서 장애가 발생된 관리객체 클래스, 관리객체 인스턴스, 그리고 장애보고에 대한 상세정보를 나타내는 매개변수('eventType', 'severity', 'backedUpStatus', 'probableCause')를 표현하고 있다. 'replaceDiscriminator'는 관리시스템으로부터 전달된 관리연산을 나타내는 외부 요인으로서 매개변수로 이벤트 보고 여부를 결정하는 'DiscriminatorConstruct' 자료형의 변수와 함께 전달된다. BDL의 EVENT 구조체는 'faultReportEvent', 'replaceDiscriminator' 등과 같은 각 외부요인을 세부적인 항목까지 엄격히 정의된 ASN.1 자료형과 함께 표현함으로써, 관리객체의 동작절차를 보다 정확하게 기술하는 기능을 제공한다.

4.2 선행조건 및 불변조건 기술

관리객체로 전달된 이벤트는 이벤트를 구성하는 매개변수의 값과 관리객체에 포함된 속성값 또는 속성의 존재여부에 따라 그 처리여부 및 처리 절차가 결정된다. 또한 관리객체가 이벤트에 대한 처리 후 반드시 만족시켜야 하는 속성값들간의 관계 혹은 무결성 제약조건과 같은 불변조건 역시 관리객체의 동작에 영향을 미치는 요소이다. 따라서 관리객체의 동작여부를 결정하는 선행조건(PRECOND) 및 불변조건(INVARIANTS)을 정형적으로 기술하는 것이 관리객체의 동적 특성을 정확히 표현하고, 프로그램 구현에 활용될 수 있는 장점이 있다.

선행조건과 불변조건은 참, 거짓을 결정할 수 있는 논리식 형태로 표현한다. 관리객체의 속성 자료형이 ASN.1을 사용하므로 조건식에 포함되는 변수들의 자료형은 ASN.1에서 지원하는 자료형과 각 자료형에 대한 연산자들이 포함되어야 한다. 외부 이벤트를 처리하기 위한 선행조건의 예를 살펴보면 다음과 같다.

PRECOND:

```
administrativeState == unlocked AND
operationalState == enabled AND
availabilityStatus != offDuty AND
availabilityStatus != logFull;
```

(그림 6) 선행조건 기술의 예

(그림 6)의 'administrativeState', 'operationalState', 'availabilityStatus'는 'log' 관리객체 클래스의 속성들로서 현재 관리객체 동작여부와 사용허가 여부 등의 상태정보를 기술하는 기능을 수행하고 있다. 따라서 관리객체는 입력으로 주어지는 이벤트를 처리하기 위해서는 관리객체의 현재 상태가 이와 같은 선행조건을 먼저 만족시켜야만 한다.

불변조건으로는 관리객체의 속성값이 가질 수 있는 범위 또는 속성값들간의 종속관계 등이 표현된다. 관리객체에 대한 이벤트 처리과정 전후 뿐만 아니라, 관리객체가 생성되어 소멸될 때까지 항상 만족시켜야 하는 제약사항 등도 불변조건에 포함될 수 있으며 관리객체의 무결성(integrity)을 유지하는 조건으로 사용될 수 있다.

INVARIANTS:

```
currentLogSize >= 0 AND
currentLogSize <= maxLogSize AND
currentLogSize = numberOfRecords * logRecordSize;
```

(그림 7) 불변조건 기술의 예

(그림 7)은 로그 데이터를 저장하는 관리객체인 'log' 관리객체 클래스에 포함된 관리속성간의 불변조건을 기술하고 있다. 속성 'currentLogSize'는 현재 'log' 관리객체에 저장된 로그 데이터의 크기를, 'maxLogSize'는 현재의 'log' 관리객체가 저장할 수 있는 로그 데이터의 최대크기를 나타내고 있다. 속성 'numberOfRecords'는 현재 'log' 관리객체에 저장되어 있는 로그 데이터의 논리적인 개수를 의미한다. 이처럼 불변조건은 관리객체에 대한 모든 동작과정들이 관리객체의 무결성을 보장할 수 있도록 관리객체에 대한 연산 수행 전과 수행후에 그에 대한 만족여부가 점검된다. 따라서 통신망 관리시스템 개발자가 응용 프로그램에서 각 관리객체에 대한 무결성 점검을 직접 구현하지 않고, 불변조건을 확인할 수 있는 코드를 자동 생성함으로써

개발과정에 활용될 수 있는 장점을 제공한다.

4.3 동작과정 기술

관리객체는 통신망 관리 측면에서 필요한 정보를 저장, 관리하며, 관리자에 의해 전달되는 관리 연산을 수행하거나 관리객체로 전달되는 외부 이벤트에 대한 처리 절차를 수행함으로써 통신망 관리에 필요한 기능을 제공한다. 따라서 관리객체의 동작과정은 4.1, 4.2에서 설명된 바와 같이 관리객체로 전달되는 외부요인(관리 연산, 통지)이나 관리객체 내부의 속성값, 조건부 패키지의 포함여부 등을 이용한 ECA 규칙을 기반으로 작성된다. 관리객체의 동작 구조는 사건위주(event-driven) 방식으로서 관리객체에 대한 사건(EVENT)이 발생되면, 발생한 사건의 처리여부를 결정(PRECOND)하고 사건에 대한 적절한 조치(ACTION)를 수행하는 구조를 가진다. 이밖에도 수행한 조치에 대한 결과의 점검을 위해 처리 전과 후에 만족되어야 하는 추가적인 조건(INVARIANTS)도 기술한다.

모든 관리객체에 대한 동작 과정을 정형적으로 기술하는 방법의 설계는 매우 힘든 작업이므로, 본 논문에서 제시하는 관리객체 동작 과정에 대한 기술은 일반적인 관리객체의 동작절차 기술에 공통적으로 적용될 수 있는 제어구조와 외부요인, 선행 및 불변조건을 이용한 프로그래밍 언어 구조를 갖는다. 먼저, 관리객체의 동작과정을 기술하기 위해 관리객체 동작절차를 결정하는 입력 이벤트의 매개변수값과 관리객체의 속성값 등의 비교연산 기능이 제공되어야 한다. 또한 관리객체 속성값의 변경과 통지의 생성 기능 역시 관리객체 동작과정을 기술하는 주요부분이다. 관리객체 클래스에 조건부 패키지가 정의되어 있는 경우, 관리객체의 동작과정이 조건부 패키지의 포함여부에 따라 달라지므로 포함여부를 판단하는 연산자가 필수적이다. 그리고, 관리객체로 전달된 외부요인의 처리결과에 따라 통지를 발생(emit)시키거나 관리객체의 동작을 종료(abort)시키는 구조체도 지원한다. 관리객체 동작에 영향을 미치는 관리연산과 통지에 대한 동작 과정을 정형적으로 기술한 예를 들면 다음과 같다.

PROCEDURE:

```
if ( EXISTS durationPackage AND
currentTime VALID IN [ startTime, end
Time ] ) then
```

```

if ( managedObjectClass == equipment
    Fault AND
    eventType == powerFailure AND
    severity == critical ) then
    emit equipmentAlarm notification;
endif;
endif;
(1) faultReportEvent 외부 요인에 대한 동작과정 예
    
```

PROCEDURE:

```

discriminatorConstruct = inDiscriminatorConstruct;
emit attributeValueChange notification;
(2) replaceDiscriminator 외부 요인에 대한 동작과정 예
    
```

(그림 8) 동작과정 기술의 예

(그림 8)은 관리객체에 입력된 외부입력의 종류와 이벤트를 발생시킨 관리객체 클래스, 그리고 매개변수 값 등을 점검하여 관리객체가 처리할 수 있는 이벤트 인지를 확인하는 과정을 보여주고 있다. 관리객체의 동작과정 중 외부 관리객체나 관리시스템에 통보해야 할 사항은 통지를 생성시켜서 'destination' 속성에 저장된 관리객체로 전송한다. 그림 8 (1)의 첫번째 문장은 관리객체 클래스가 정의하고 있는 조건부 패키지가 관리객체에 포함되었는지를 확인하고 그에 따른 제약 사항의 기술의 예를 보이고 있다. 조건부 패키지가 포함되는 지에 대한 실제적인 확인기능은 'top' 관리객체 클래스의 'Packages' 속성값의 점검을 통해 판단할 수 있다. 'currentTime'은 관리객체에 의해 이벤트가 처리되는 시각을 나타내는 속성으로, 관리객체를 저장하고 처리하는 시스템에서 그 값을 제공할 수 있다.

그러나, 각각의 관리객체마다 다른 동작특성을 가지고 있기 때문에 관리객체 동적 특성을 모두 정형적으로 기술하기는 매우 어려운 일이다. 따라서 본 논문에서는 ITU-T 표준권고안에 정의된 관리객체 속성이 가질 수 있는 자료형을 기반으로, 모든 관리객체의 공통적으로 적용될 수 있는 관리객체 속성값과 입력 매개변수값 등을 비교할 수 있는 판단문, 관리객체 속성값의 변경, 통지의 발생 기능만 정형적으로 기술하고, 관리객체에 따라 정형적으로 표현이 불가능한 동적 속성에 대하여는 텍스트 형태로 기술할 수 있도록 구문을 정의하였다(부록 참조).

4.4 관리객체의 동적 특성 기술 예

여기서는 로그 관리 기능(Log Control Function)에 대한 X.735 권고안에 기술된 'log' 관리객체의 동적 특성을 본 논문에서 제안된 BDL을 이용하여 기술한 예를 살펴본다. 'log' 관리객체는 외부로부터 전달되는 이벤트 보고나 로컬 시스템에서 발생된 통지의 저장 기능을 수행한다. 그러나, 관리객체의 동작상태를 나타내는 'administrativeState', 'operationalState', 'availabilityStatus' 속성값이나 'discriminatorConstruct' 속성값에 의해 관리객체로 전달된 통지나 이벤트 보고를 로그에 저장될 것인지 결정된다. 또한 로그 관리객체의 동작 과정중 예외적인 상황의 처리를 위한 'finiteLogSizePackage', 'logAlarmPackage' 조건부 패키지, 그리고 시간지원 특성을 나타내는 'availabilityStatusPackage', 'duration', 'dailyScheduling', 'weeklyScheduling', 'externalScheduling' 조건부 패키지들이 모두 로그 관리객체의 동적 특성을 결정하는 요인들이다. 그러나, 현재 권고안에는 이러한 복잡한 로그 관리객체의 동작 과정을 다음과 같이 표현하고 있다(그림 9).

```

log MANAGED OBJECT CLASS
DERIVED FROM top;
CHARACTERIZED BY
logPackage PACKAGE
BEHAVIOUR
logBehaviour BEHAVIOUR
DEFINED AS "This managed object is used to store incoming event reports and local system notifications. Additional details are defined in CCITT. X. 735/ISO /IEC 10164-6.";;
ATTRIBUTES
logID GET,
discriminatorConstruct GET-REPLACE,
administrativeState GET-REPLACE,
operationalState GET,
availabilityStatus PERMITTED VALUES Attribute-ASN1Module.LogAvailability
REQUIRED VALUES Attribute-ASN1Module.UnscheduledLogAvailability
GET,
logFullAction GET-REPLACE;
...
    
```

(그림 9) X.735에 기술된 log 관리객체 클래스의 동작 특성

(그림 10)은 BDL을 이용한 로그 관리객체의 동적 특

성중 다른 객체로부터 전달된 통지에 대한 처리과정과 통신망 관리자로부터 전달된 속성값 변경에 대한 관리 명령 수행 절차를 나타내고 있다.

```

logPackage PACKAGE
BEHAVIOUR
logBehaviour BEHAVIOUR
DEFINED AS
EVENT: internalNotification /* 내부 통지 이벤트
                               기술 */

notification: smi2Notificaiton
notificationID: NotificationIdentifier;
PRECOND: /* 선행조건 기술 */
administrativeState == unlocked AND
operationalState == enabled AND
availabilityStatus != offDuty AND
availabilityStatus != logFull;
PROCEDURE: /* 동작절차 기술 */
logRecordSize : INTEGER;
"set the logRecordSize according to data logged";
if ( NOT ( FILTER ( discriminatorConstruct,
notification ) ) )
    abort: /*식별자구조에 통지 비포함시 철회 */
endif;
if ( NOT ( EXISTS durationPackage AND
currentTime VALID IN [ startTime,
endTime ] ) ) then
    abort: /* 스케줄링 시간 이외에 통지 철회 */
endif;
if ( EXISTS finiteLogSizePackage ) then /*
Log의 크기가 지정 */
"process the notification suitable to log
record structure, add modified notification
information to log storage";
numberOfRecords = numberOfRecords + 1;
currentLogSize = currentLogSize + logRecord
Size;
if ( currentLogSize >= maxLogSize ) then
/* 최대 log 크기 초과시 */
if ( logFullAction == halt ) then
    availabilityStatus = logFull;
endif;
if ( logFullAction == wrap ) then
"delete the oldest log record and
allocate it to new log record"
endif;
end if;
endif;

EVENT: replaceDiscriminator /* 식별자 속성변경
                               이벤트 기술 */
    
```

```

inDiscriminatorConstruct: DiscriminatorConstruct;
PRECOND:
administrativeState == unlocked AND
operationalState == enabled;
PROCEDURE:
discriminatorConstruct = inDiscriminatorConstruct;
/* 새로운 입력 식별자구조로 치환 */
emit attributeValueChange notification;
...
    
```

(그림 10) BDL을 이용한 log 관리객체 클래스의 동작 특성 기술

(그림 10)의 'internalNotification' 외부요인에 대한 처리 절차중 'FILTER' 함수는 입력으로 주어진 'notification'의 속성값을 현재 'log' 관리객체의 'discriminatorConstruct'가 가지는 필터항목과 비교하여 'log' 관리객체에 저장될 수 있는 통지인지 점검하는 기능을 제공한다. 그리고 위의 예제에서는 시간지원 속성중 'duration' 조건부 패키지에 대한 포함여부 확인 기능만을 보여주고 있으나, 필요에 따라 다른 시간지원 조건부 패키지에 대한 포함여부 점검 및 동작과정의 추가도 가능하다. 'logFullAction' 속성값은 'finiteLogSizePackage' 조건부 패키지가 포함될 때 정의되는 속성으로 'log' 관리객체에 더 이상 로그 정보를 저장할 수 없을 때 취하는 조치를 정의하고 있다. 'logFullAction' 속성값이 'halt'인 경우는 'log' 관리객체의 동작이 'availabilityStatus' 속성값을 'logFull'로 지정함으로써 'log' 관리객체의 동작을 정지시키며, 'wrap'인 경우에는 'log' 로그 관리객체에 저장된 가장 오래된 로그 정보를 지우고 그 영역에 새로운 로그 정보를 저장한다. 'replaceDiscriminator' 외부요인은 통신망 관리자에 의해 입력으로 전달된 'DiscriminatorConstruct' 자료형의 새로운 로그 필터용 구조체를 'log' 관리객체의 'discriminatorConstruct' 속성값에 새로 저장하는 기능을 수행함으로써, 관리연산 수행후에 'log' 관리객체에 새로운 타입의 로그 정보가 저장되도록 한다. 한편, 'internalNotification', 'replaceDiscriminator' 외부요인 모두 'PRECOND'에 의해 지정되는 조건이 만족된 경우에만 위에서 기술된 동작 절차가 수행된다.

위와 같이 본 논문에서 제시한 통신망 관리객체의 동적 특성을 결정하는 외부요인과 조건들, 그리고 능동적,

시간지원 특성에 기반을 둔 동작 절차를 BDL을 이용하여 기술하는 방법론이, 비록 모든 관리객체들에 대한 세부적인 각각의 동작 과정을 기술하지는 못하지만, 지금까지 자연어를 이용하여 비정형적으로 기술되었던 통신망 관리객체의 동적 특성을 보다 체계적이고 정형적으로 기술하게 되는 장점을 제공함을 볼 수 있다.

5. 결론 및 향후 연구방향

다양한 통신 프로토콜과 이질적인 컴퓨터 시스템으로 구성되는 현재의 통신망을 효과적으로 관리하기 위해 ITU-T, ISO 국제표준기구들은 개방형 통신모델, 관리정보모델, 기능모델 등을 정의하기 위한 표준권고안들을 제정하였다. 이들 중 관리정보모델은 통신망 관리자에게 일관적인 관리 인터페이스의 제공을 위하여 통신망 관리에 필요한 통신망 구성요소의 특성에 대한 모델링과 구조 등을 기술하고 있다. 통신망 구성요소는 관리객체의 형태로 표현되며, 관리객체는 GDMO 표현법을 이용하여 통신망 관리에 필요한 물리적 또는 논리적인 통신망 구성요소의 정적, 그리고 동적인 특성을 체계적으로 표현한다. 그러나, GDMO 표현법중 관리객체 동적 특성의 기술을 정형적인 방법 대신 자연어를 이용함으로써 관리객체의 동작 절차와 기능을 표현하는데 한계가 있는 문제점이 있다.

본 논문에서는 관리객체의 동적인 특성을 체계적이고 정형적으로 기술하여 관리객체가 수행하는 기능을 위한 정형적인 기술방법에 대해 기술하였다. 관리객체의 기능이 관리객체 클래스마다 다른 특성으로 인해 모든 관리객체에 대한 동적 특성을 완전하게 기술할 수 있는 일반적인 방법을 설계하는 작업이 용이하지 않지만, 본 논문에서는 관리객체에 공통적으로 적용될 수 있는 동적 특성을 기술하기 위한 외부요인, 선행 및 불변조건, 그리고 동작절차를 결정하는 논리조건과 능동적 특성 외에 지금까지 관리객체의 동적 특성에서 고려되지 않았던 시간지원 속성들을 표준권고안을 준용하여 정의하고, 이를 프로그래밍 언어 형태를 이용하여 기술할 수 있도록 BDL 문법을 정의하였다. 제시된 정형적 기술방법은 통신망 관리 프로그램의 자동생성과 시험에 활용될 수 있을 것이며, 제안된 정형적 기술언어의 구조와 기능을 보완하고, 기술언어 각 구성요소의 의미에 대한 정형화 연구를 진행할 계획이다.

부 록

관리객체 동적 특성 기술언어(Behaviour Description Language) 문법 (EBNF 형식)

```
Behaviour-Description ::=
{ 'EVENT:'
  Attribute-List ':' Type-Name
  Attribute-List ':' Type-Name { ';' Attribute-List ':' Type-Name } * ';'
  'PRECOND:'
  [ Condition-List ] ';'
  [ 'INVARIANTS:' ]
  [ Condition-List ] ';'
  'PROCEDURE:'
  [ Attribute-List ':' Type-Name { ';' Attribute-List ':' Type-Name } * ';' ]
  Statement-List ';'
} *

Attribute-List ::= Attribute { ',' Attribute } *
Statement-List ::= Statement { ',' Statement } *
Condition-List ::= Condition-List Logical-Op Condition
                  | Condition
                  | 'NOT' '(' Condition ')'

Condition ::= Condition Relational-Op Primary-Exp
            | Condition Set-Relational-Op Primary-Set-Exp
            | Primary-Exp
            | Primary-Set-Exp
            | String-Exp
            | Temporal-Exp
            | 'EXISTS' Attribute
            | 'FILTER' '(' Attribute ',' Attribute ')'

Primary-Exp ::= Primary-Exp Numeric-Op Exp
              | Exp

Primary-Set-Exp ::= Primary-Set-Exp Set-Op Set-Exp
                  | Set-Exp

Set-Exp ::= Set-Exp Set-Op Primitive-Set-Exp
          | Primitive-Set-Exp

String-Exp ::= Primary-String-Exp String-Op Primary-String-Exp
             | String-Exp

Temporal-Exp ::= Attribute Temporal-Op Temporal-Interval
               | 'currentTime' Temporal-Op Temporal-Interval

Temporal-Interval ::= '[' Time-Point ',' Time-Point ']'
                  | Attribute-ASNIModule.defaultIntervalsOfDay
                  | Attribute-ASNIModule.defaultWeekMask

Statement ::= Assignment-Statement
```

```

    | Conditional-Statement
    | Notification-Statement
    | Text-Statement
    | 'abort'
Logical-Op ::= 'AND' | 'OR'
Relational-Op ::= '=' | '<' | '>' | '<=' | '>=' | '!='
Numeric-Op ::= '+' | '-' | '*' | '/'
Set-Relational-Op ::= 'SUBSET OF'
                    | 'SUPERSET OF'
                    | 'NOT NULL SET INTERSECTION'
Set-Op ::= 'UNION'
          | 'INTERSECTION'
          | 'DIFFERENCE'
String-Op ::= 'EQUAL'
             | 'STARTS WITH'
             | 'ENDS WITH'
             | 'INCLUDES'
             | 'CHANGES'
Temporal-Op ::= 'VALID IN'
               | 'NOT VALID IN'
Type-Name ::= { x | x is attribute type in Attribute-
                ASNModule }
Exp ::= Number
      | Attribute
      | '(' Primary-Exp ')'
Primitive-Set-Exp ::= { x | x is ANY type }
                   | Attribute
                   | '(' Primary-Set-Exp ')'
Primary-String-Exp ::= Attribute-ASNModule. Additional
                     Text
                     | Attribute
                     | '(' String-Exp ')'
Number ::= INTEGER
         | REAL
Time-Point ::= Attribute-ASNModule.GeneralizedTime
Attribute ::= Identifier
Assignment-Statement ::= Attribute '=' Primary-Exp
                      | Attribute '=' Primary-Set-Exp
                      | Attribute '=' String-Exp
Conditional-Statement ::= 'if' '(' Condition ')' 'then'
Statement-List 'endif' ';'
Notification-Statement ::= 'emit' Attribute 'notification' ';'
Text-Statement ::= ''
Attribute-ASNModule.AdditionalText ::= ''

```

참 고 문 헌

- [1] ISO International Standard 7498 Information Processing Systems - Open Systems Interconnection - Basic Reference Model Part 4: Management Framework 1991.
- [2] ISO International Standard 8824 Information Processing Systems - Open Systems Interconnection - Specification of ASN.1(Abtract Syntax Notation One) 1987.
- [3] ITU-T X.711|ISO DIS 9595: "Information Processing Systems - Open Systems Interconnection - Common Management Information Service Definition," Geneva, 1990.
- [4] ITU-T X.712|ISO DIS 9596: "Information Processing Systems - Open Systems Interconnection - Common Management Information Protocol," Geneva, 1991.
- [5] ITU-T X.701|ISO DIS 10040: "Information Processing Systems - Open Systems Interconnection - System Management Overview," Geneva, 1991.
- [6] ITU-T X.720|ISO DIS 10165-1: "Information Processing Systems - Open Systems Interconnection - System Management," Geneva.
- [7] ITU-T X.721|ISO DIS 10165-2: "Information Processing Systems - Open Systems Interconnection - Management Information Services - Structure of Management Information Part 2: Definition of Management Information," Geneva.
- [8] ITU-T X.722|ISO DIS 10165-4: "Information Processing Systems - Open Systems Interconnection - Management Information Services - Structure of Management Information Part 4: Guidelines for the Description of Managed Objects," Geneva.
- [9] ITU-T X.723|ISO DIS 10165-5: "Information Processing Systems - Open Systems Interconnection - Management Information Services - Structure of Management Information Part 5: Generic Management Information."
- [10] ITU-T X.725: "Information Processing Systems - Open Systems Interconnection - Structure of Information: General Relationship Model."

[11] ITU-T X.731|ISO DIS 10164-2: "Information Technology - Open Systems Interconnection - System Management - State Management Function."

[12] ITU-T X.734|ISO DIS 10164-5: "Information Technology - Open Systems Interconnection - System Management - Event Report Management Function."

[13] James Rumbaugh, Michael Blaha, William Premerlani, Frederick Eddy, William Lorensen, Object-Oriented Modeling and Design, Prentice-Hall, 1991.

[14] J. Derrick, P. F. Linington, S. J. Thompson, "Formal Description Techniques for Object Management," Proceedings of the 4th International Symposium on Integrated Network Management, 1995.

[15] Masum X. Hasan, "An Active Temporal Model for Network Management Databases," Proceedings of the 4th International Symposium on Integrated Network Management, 1995.

[16] Oliver Festor, Georg Zornlein, "Formal Description of Managed Object Behavior-A Rule Based Approach," Proceedings of the IFIP TC6/ WG6.6 3rd International Symposium on Integrated Network Management, 1993.

[17] Yechiam Yemini, Alex Dupuy, Shmuel Kliger, Shaula Yemini, "Semantic Modeling of Managed Information," Network Management and Control, Plenum Press, 1994.

[18] Alexander Clemm, "Incorporating Relationships into OSI Management Information," 2nd IEEE Network Management and Control Workshop, 1993.

[19] Alexander Clemm, "Incorporating Relationships into OSI Management Information," Network Management and Control, Vol.2, Plenum Press, 1994.

[20] J. Keller, "An Extension of GDMO for Formalizing Managed Objects Behaviour," Proceedings of the IFIP TC6 8th International Conference on Formal Description Techniques, 1995.

[21] Morris Sloman, Network and Distributed Sys-

tems Management, Addison-Wesley, 1994.

[22] Adrian. Tang et al. Open Networking with OSI, Prentice-Hall, 1992.

[23] Divakara K. Udupa, Network Management Systems Essentials, McGraw-Hill, 1996.



최은복

e-mail : eunbog@chonnam.chonnam.ac.kr
 1992년 전남대학교 전산학과 졸업 (이학사)
 1996년 전남대학교 전산학과 졸업 (이학석사)
 1997~현재 전남대학교 전산학과 박사과정 수료

관심분야 : 통신망관리, 정보보안, 멀티미디어시스템 등



이형효

e-mail : hlee@athena.chonnam.ac.kr
 1987년 전남대학교 전산학과 졸업 (이학사)
 1989년 한국과학기술원 전산학과 졸업(공학석사)
 1990년~1992년 삼보컴퓨터 기술 연구소

1993년~1997년 한국통신 연구개발원
 1995년 정보처리기술사(전자계산기응용)
 1997년~현재 전남대학교 전산학과 박사과정 수료
 관심분야 : 통신망관리, 정보보안, 객체지향시스템 등



노봉남

e-mail : bongnam@chonnam.chonnam.ac.kr
 1978년 전남대학교 수학교육과 졸업(이학사)
 1982년 한국과학기술원 전산학과 (공학석사)
 1994년 전북대학교 대학원 전산 통계학과(이학박사)

1983년~현재 전남대학교 전산학과 교수
 관심분야 : 객체지향시스템, 통신망관리, 정보보안, 컴퓨터와 정보사회 등