

동적 시간지원 특성을 지원하는 망관리 객체의 정형적 모델링

최 은 복[†] · 이 형 효^{††} · 노 봉 남^{†††}

요 약

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 구성요소중 관리객체에 동작에 영향을 미치는 능동적 요인들과 동작절차 및 그에 따른 동시 발생 등의 과정들이 'BEHAVIOUR' 템플릿내에 자연어를 이용하여 매우 포괄적이며 간략히 기술되어 있어, 관리객체의 동작 조건이나 절차, 결과 등이 명확히 표현되지 못하는 문제점을 가지고 있다. 또한 관리객체의 동적 특성은 능동적 특성 외에 시간지원 특성에 따라 동작 절차가 달라질 수 있다. 반복적이고 주기적인 정적인 시간 개념 뿐만 아니라 다른 관리자의 속성에 따라 사용권한이 결정되거나 특정기간 동안에만 권한이 부여되는 동적인 시간개념이 첨가된다면 관리객체의 능동적 특성이 더욱 강화될 수 있다

본 논문에서는 정적인 시간관리객체의 활동을 제어하는 표준권고안의 스케줄링 관리객체에 동적인 특성을 추가하였으며 관리객체 기술의 표준인 GDMO 표현법에 의해 동적특성을 지원하는 스케줄링 관리객체를 기술하였다. 또한 관리객체의 동적 특성의 기술을 위해 새롭게 정의한, 동적 시간지원 기능이 추가된 BDL을 이용하여 동작절차를 기술하였다. 그리고 동적 시간지원 특성을 지원하는 관리객체를 대리자 관리 기능 모델에 적용하여 체계적이고 정형적인 기술방법임을 제시한다.

A Formal Modeling of Managed Object Behaviour with Dynamic Temporal Properties

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

ABSTRACT

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. The current recommendations provide the formal mechanism for the structural parts of managed objects such as managed object class and attributes. 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. Also, the behaviour of managed objects is affected by their temporal and active properties. While the temporal properties representing periodic or repetitive intervals are to describe managed objects' behaviour in rather strict way, it will be more powerful if more dynamic temporal properties determined by external conditions are added to managed objects. In this paper, we added dynamic features to scheduling managed objects, and described, in GDMO, scheduling managed objects that support dynamic features. We also described behaviour of managed objects in newly defined BDL that has dynamic temporal properties. This paper showed that dynamic temporal managed objects provide a systematic and formal method in agent management function model.

† 준 회원 : 전남대학교 대학원 전산학과
†† 정 회원 : 전남대학교 대학원 전산학과
††† 종신회원 : 전남대학교 전산학과 교수
논문접수 : 1999년 7월 3일, 심사완료 : 1999년 9월 10일

1. 서 론

새로운 통신기술과 다양한 컴퓨터 네트워크의 결합으로 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 자원들의 속성(attribute)과 동적 특성(behaviour)에 대한 추상화된 표현방법, 그리고 망 구성요소들에 대한 관리기능을 포괄적으로 규정하고 있다. 따라서 이러한 표준 권고안들은 근거로 구현된 통신망 관리시스템은 이질적인 통신망을 포함하는 전체 통신망의 상호운용성(interoperability) 제공과 통합적인 망관리 기능의 수행을 가능하게 한다[7, 14].

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[15].

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 관리시스템에게 관리객체 내부에서 관리상에 주요한 사항이 발생했음을 알리는 통지(notification)에 대해서 자세히 기술하고 있다. 그러나 현재의 GDMO 표현은 관리객체의 구조와 속성등을 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 있

다. 또한, 관리객체의 동적 특성은 능동적 특성 외에 시간지원 특성에 따라 동작 절차가 달라질 수 있다. 이러한 관리객체의 동작 절차에 통신망 관리 목적이나 관리객체의 특성에 따라 지정된 기간 또는 반복적인 작동 주기를 가지고 동작되는 정적인 시간개념과, 다른 관리자의 속성에 따라 사용권한이 결정되거나 특정 기간 동안에만 권한이 부여되는 동적인 시간개념을 첨가하여 관리객체의 능동적 특성을 더욱 강화할 필요가 있다.

본 논문에서는 정적인 시간관리객체의 활동을 제어하는 표준권고안의 스케줄링 관리객체에 동적인 특성을 추가하였으며 관리객체 기술의 표준인 GDMO 표현법에 의해 동적특성을 지원하는 스케줄링 관리객체를 기술하였다. 또한 통신망 관리에 대한 표준권고안과 관리객체의 동적 특성의 기술을 위해 새롭게 정의한, 동적 시간지원 기능이 추가된 BDL(Behaviour Description Language for Managed Objects)을 이용하여 동작절차를 기술하였다[부록 참조]. 그리고 동적 시간지원 특성을 지원하는 관리객체를 대리자 관리 기능 모델에 적용하여 체계적이고 정형적인 기술방법임을 제시한다.

이러한 프로그래밍 언어 형태의 정형적 기술방법은 관리객체의 동작 특성을 결정하는 요인들과 동작 과정을 명확히 기술함으로써 관리객체의 설계자나 프로그래머들이 관리객체의 동적 특성을 표현하고 이해하는데 도움이 되며, 관리객체의 구현단계에서 관리객체의 주요 기능에 대한 프로그램의 자동생성에 활용될 수 있는 장점을 제공한다.

2. OSI 관리 모델

관리객체는 통신망을 구성하는 다양한 구성요소들에 대하여 객체지향개념을 기반으로 설계되었으며 표준화된 관리 인터페이스 기능을 제공하고 있다[18, 20]. 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 관리자에게 제공하는 기능을 수행한다[16]. 그리고 유사한 데이터와 연산을 가지는 관리객체들의 그룹을 관리객체 클래스로 정의된다. ITU-T X.722 권고안에서는 객체지향

개념의 특성인 캡슐화, 정보 은닉성, 상속성 등을 활용하여 관리객체 클래스를 정의함으로써 표준화된 통신망 관리시스템의 설계와 효과적인 개발을 지원하고 있다[18]. GDMO 표현법에서는 MANAGED OBJECT CLASS, PACKAGE, ATTRIBUTE, ATTRIBUTE GROUP, ACTION, PARAMETER, NAME BINDING, BEHAVIOUR, NOTIFICATION 등 9개의 템플릿을 이용하여 관리객체 클래스의 정적 및 동적 특성을 기술하고 있다[19].

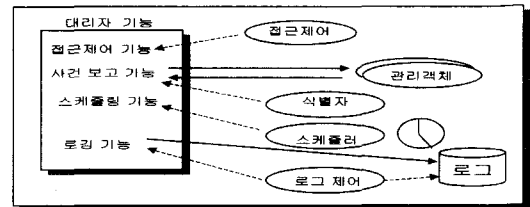
그러나, 관리객체 클래스를 정의하기 위하여 사용되는 PACKAGE, ATTRIBUTE, ACTION NOTIFICATION 등과 같은 템플릿들의 동작특성을 기술하는 BEHAVIOUR 템플릿이 위와 같은 특성을 표현할 수 있는 체계나 구분 등이 현재의 권고안에는 정의되어 있지 않아서, 관리객체의 동적 특성을 체계적이면서 명확하게 표현할 수 없는 문제가 있다. 그리고, 관리객체의 동적 특성 기술의 기준과 범위가 존재하지 않아, 관리객체마다 동적 특성을 기술하는 방법과 명세정도가 달라서 통신망 관리시스템 개발자에게 시스템 개발에 필요한 정보를 충분히 제공하지 못하였다.

또한, 관리객체의 동적 특성은 능동적 특성 외에 시간지원 특성에 따라 동작 절차가 달라질 수 있다. 관리객체의 시간지원 동작 절차는 통신망 관리 목적이나 관리객체의 특성에 따라 정적인 시간개념인 지정된 기간 또는 반복적인 작동 주기를 가지고 동작될 수 있다. 이러한 정적인 시간 개념 뿐만 아니라 다른 관리자의 속성에 따라 사용권한이 결정되거나 특정기간 동안에만 권한이 부여되는 동적인 시간개념이 첨가된다면 관리객체의 능동적 특성이 더욱 강화될 수 있다

2.1 대리자 관리 기능 모델

(그림 1)의 대리자 관리 기능 모델에서 대리자 관리 객체는 망 관리 서비스를 제공할 뿐만 아니라 관리 객체 인터페이스와 개방형 통신 인터페이스간의 매핑을 제공한다. 또한 대리자에서는 수행되는 연산을 선택적으로 필터링하는 메커니즘과 통지를 통해 발생하는 데이터의 흐름을 제어하는 기능을 제공한다. 국제표준안에서 제공하는 대리자의 기능에는 접근제어, 사건 보고, 스케줄링, 그리고 로깅 기능 등이 있다[23]. 관리자 와 관리객체에 대한 접근 권한의 허가 여부를 결정하

는 접근제어 기능과 관리객체에 대한 특정 사건이 발생하였을 때 관리자에게 통지하는 사건 보고 기능, 주기적이고 반복적인 스케줄링 계획에 따라 관리객체의 활동을 제어하는 스케줄링 기능 그리고 사건이나 통지, 관리객체 접근 등 전반적인 내용에 대한 감사 추적을 위한 로깅 기능 등이 있다.



(그림 1) 대리자 관리 기능 모델

2.2 시간지원 속성

관리객체의 동적 특성은 능동적 특성 외에 시간지원 특성에 따라 동작 절차가 달라질 수 있다. 관리객체의 동작은 통신망 관리 목적이나 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 운용조건을 가지는 관리객체의 동적 특성을 정확히 기술할 수 있다. 예를 들어, 'discriminator' 관리객체 클래스의 경우 'discriminator' 관리객체에 전달된 사건은 'discriminator' 속성에 정의된 필터에 의해 사건 보고 생성 여부를 결정한다. 이때 사건 보고 발생을 결정하는 또 다른 요인이 스케줄링 패키지로서 시간제약 조건을 포함하고 있다. 스케줄링 패키지는 복잡한 사건 보고 주기(period)의 관리를 위해 정의된 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 대

한 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[25].

3. 관리객체의 동적 특성 모델링의 확장

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 외부요인에 의해 그 상태가 변화되는 과정을 기술한다[12].

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 속성값들간의 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. 그리고 관리객체의 동적 특성의 기술을 위해 동적 시간지원 기능이 추가된 BDL을 정의한다. BDL은 관리객체의 동적특성을 기술하기 위한 언어로 BNF형식을 갖는다. 또한 관리자에 의해 지정되는 정적인 시간개념 뿐만 아니라 실제 관리객체에 대한 연산을 수행하는 프로세스의 활동 시점을 기준으로 처리하는 동적 시간개념을 첨가하므로서 보다 관리객체의 능동적 특성을 강화할 수 있다. 그리고 이러한 시간지원 동적 특성을 기반으로 하여 대리자 관리기능 모델에 대한 동작기술을 제

계적이고 정형적으로 기술하였다.

3.1 확장된 스케줄러 관리객체 클래스 기술

관리객체의 동적특성은 능동적 특성 외에 시간지원 특성에 따라 동작절차가 달라 질 수 있다. 관리객체의 동작은 통신망 관리 목적이나 관리객체의 특성에 따라 반복적이고 주기적인 정적인 시간 개념 뿐만 아니라 다른 관리자의 속성에 따라 사용권한이 결정되거나 특정기간 동안에만 권한이 부여되는 동적인 시간개념이 첨가된다면 매우 복잡한 운용조건을 갖는 관리객체의 동적 특성을 정확히 기술할 수 있다. 예로, 특별한 망관리 객체만 관리할 필요가 있거나 망관리 연구를 위해 동일한 사건을 발생한 관리객체를 추적 조사하는 경우와 같이 세부적인 망관리 서비스를 제공하기 위해서는 시간지원 속성이 반드시 요구되어진다.

시간표현은 year, month, week, day, hour, minute, second와 같은 시간단위로 표현되며 관리자가 대리자의 관리객체를 접근할 때 허용되는 시점이 주어지거나 특정 관리객체가 활성화되는 사건시간을 표현하는 타임스탬프, 일정한 주기적인 시간과 비주기적인 시간을 나타내는 시간간격(Interval), 일정크기의 시간을 표현하는 수행시간(Period) 그리고 이외의 부가적인 항목을 갖는 외부적인 시간 등이 있다.

시간 지원에 연관된 스케줄링 기능에는 특별한 관리객체 인스턴스 내에 명세되어 있는 시간 관리객체의 활동을 제어하여 관리객체 내에서 자체적으로 발생하는 'trigger scheduling' 인 비주기적인 스케줄링과 관리객체의 활동에 사용되는 연산의 시간구간을 지정하여 일정하게 반복적으로 제어하는 'interval scheduling' 주기적인 스케줄링으로 나뉜다. 특히, 주기적인 스케줄링에서 'duration' 패키지는 관리객체가 동작하는 시작 및 종료시간을 지정하여 관리객체의 동작을 자동으로 제어하며, 'dailyScheduling' 패키지는 하루동안에 관리객체가 동작하는 시작과 종료시간을 여러번 지정할 수 있는 기능을 제공한다. 그리고 'weeklyScheduling'은 주단위로 시작과 종료일을, 'monthlyScheduling' 월단위로 시작과 종료일을 지정할 수 있는 기능을 제공한다[25]. 본 논문에서는 이러한 시간관련 관리객체의 동적특성을 권고안에 정의되어있는 정적 시간 개념인 내부적인 스케줄링 기능 뿐만 아니라 동적 시간지원 기능을 추가한 외부적인 스케줄링인 'external-

Scheduling'을 정의함으로서 보다 강화된 동적특성을 정확하고 체계적으로 표현하고 이해하고자 한다.

3.1.1 정적 시간 속성 기술

다음은 표준권고안에 정의되어 있는 'durationPackage', 'dailySchedulingPackage', 'weeklySchedulingPackage'을 이용하여 관리객체의 동적특성을 정형적으로 기술한 동작절차의 한 예이다. 여기서 현재시간이나 현재 일, 현재 주가 지정된 시각과 스케줄링 패키지에 맞지 않으면 timeDomainViolation 통지를 발생시키고 이 연산을 종료시키는 기능을 수행한다.

```

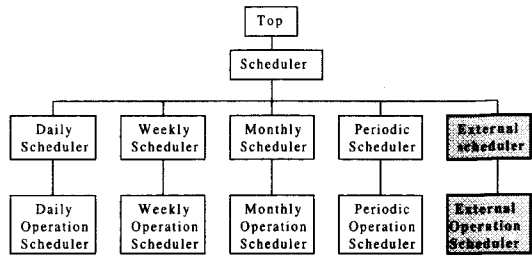
PROCEDURE:
  if ( NOT ( EXISTS durationPackage AND currentTime
            VALID IN[startTime, stopTime] ) OR
        ( EXISTS dailySchedulingPackage AND currentDay
            VALID IN[startDay, stopDay] ) OR
        ( EXISTS weeklySchedulingPackage AND currentWeek
            VALID IN[startWeek, stopWeek] ) ) then
    emit timeDomainViolation notification;
  abort;
endif;
    
```

(그림 2) 정적 시간 속성 기술의 예

3.1.2 동적 시간 속성 기술

현재 표준권고안에서는 시간 관리객체의 활동을 제어하는 스케줄링 관리 객체가 정의되어 있다. Daily Scheduler는 하루동안에 관리객체가 동작하는 시각과 종료시간을 지정하여 관리하는 객체이고 Weekly Scheduler는 일주일동안 관리객체가 동작하는 요일들과 그 요일에 대한 한번 이상의 시작 및 종료시간을 지정하여 관리하는 객체이다. Monthly Scheduler는 매달동안 관리객체가 동작하는 주를 지정하여 관리하는 객체이고 Periodic Scheduler는 지정된 시간동안 주기적으로 관리하는 스케줄러 객체이다. 이러한 스케줄링 관리객체는 관리자에 의해 시간이 지정되는 정적인 시간개념으로 동적특성을 갖는 시간지원 특성을 표현하기에는 다소 미흡하다. 그래서 다른 관리자의 속성에 따라 사용권한이 결정되거나 특정기간동안에만 권한이 부여되는 동적인 시간개념이 추가된다면 동적특성이 더욱

강화될 것이다. 그러므로 본 논문에서는 (그림 3)과 같이 X.746 권고안에 기술된 스케줄러 관리객체 클래스에 동적인 시간지원 속성을 지원할 수 있는 External Scheduler 관리객체 클래스와 External Operation Scheduler 관리객체 클래스를 정의하였으며, 이들을 표준권고안에 준용하여 GDMO 표현식으로 기술하였다.



(그림 3) 확장된 Scheduler MO Class 상속 계층구조

그리고 External Scheduler 관리객체의 필수 패키지로, 특정 관리자에 대한 관리객체의 접근모드에 따라 해당 관리자의 접근여부가 결정되는 dependencySchedulingPackage패키지와 관리자가 관리객체에 접근요청을 시작하는 시점을 기준으로 하여 종료시점을 부여하는 timestampSchedulingPackage를 정의하였다.

• External Scheduler and ExternalOperationScheduler

External Scheduler와 External Operation Scheduler 관리객체 클래스를 권고안에 근거하여 GDMO 표현식으로 기술하면 다음과 같다. ExternalScheduler 클래스는 표준 권고안에 기술된 scheduler 클래스로부터 속성을 상속받으며 필수적으로 수행하는 dependencySchedulingPackage와 timestampSchedulingPackage를 갖는다.

```

ExternalScheduler MANAGED OBJECT CLASS
  DERIVED FROM scheduler;
  CHARACTERIZED BY
    dependencySchedulingPackage;
    timestampSchedulingPackage;
  REGISTERED AS {schedMo 11};
    
```

(그림 4) ExternalScheduler 관리객체 클래스

```

ExternalOperationScheduler MANAGED OBJECT CLASS
DERIVED FROM ExternalScheduler;
CHARACTERIZED BY
    operationSchedulingPackage;
CONDITIONAL PACKAGES
    operationNotificationPackage PRESENT IF "the result
of the operation need to be reported or the operation
performed is a GET operation";
REGISTERED AS {schedMo 12};
    
```

(그림 5) ExternalOperationScheduler 관리객체 클래스

• **dependencySchedulingPackage**

특정 관리자에 대한 관리객체의 접근모드에 따라 해당 관리자의 접근여부가 결정되는 dependencySchedulingPackage를 정의하였다. 여기서 적용되는 모드에는 특정관리자가 관리객체에 대해 접근모드를 수행하기전에만 수행할 수 있는 'UNLESS' 모드와 특정 관리자가 관리객체에 대해 접근모드를 수행하지 않는 그 이외의 시기에만 수행할 수 있는 'WHENEVERNOT' 모드가 있으며, 이와 상반된 개념으로, 특정 관리자가 관리객체에 대해 접근모드를 수행할때만 접근할 수 있는 'WHENEVER'모드와 특정관리자가 관리객체에 대해 접근모드를 최초로 수행할 때만 접근할 수 있는 'ASLONGAS' 모드가 있다[1].

이에 해당하는 규칙을 정의하면 다음과 같다.

<정의 1> dependencyScheduling 규칙

```

(mi, ai, moi, p) <dependencyMode> (mj, aj, moj, p)
단, mi, mj ∈ M : 관리자 (i ≠ j), ai ∈ A : 대리자,
moi ⊆ MO : 관리객체, p ∈ P : ACCESS_MODE
    
```

한 예로 다음 (m₂, a₁, {o₁}, {M-GET}) WHENEVERNOT (m₁, a₁, {o₁}, {M_GET}) 규칙의 경우 관리자 m₂는 관리자 m₁이 대리자 a₁의 관리객체 o₁에 대해 M-GET 연산을 수행할 수 있음을 의미한다.

이를 근거로 한 dependencySchedulingPackage패키지와 ManagedObjectMode 속성그룹 그리고 DependencyMode 속성을 기술하면 다음과 같다.(그림 6, 7)

```

dependencySchedulingPackage PACKAGE
BEHAVIOUR dependencySchedulingBehaviour BEHAVIOUR
DEFINED AS
EVENT : checkedDependencyMode
    delegatedManagedObject : ManagedObjectMode,
    dependencyMode : DependencyMode,
    sourceManagedObject : ManagedObjectMode;
PRECOND :
    administrativeState == unlocked AND
    operationalState == enabled AND
    availabilityStatus != Offduty;
PROCEDURE :
    if ( NOT( (delegatedManagedObject UNLESS
        sourceManagedObject) OR
        (delegatedManagedObject WHENEVERNOT
        sourceManagedObject) OR
        (delegatedManagedObject ASLONGAS
        sourceManagedObject) OR
        (delegatedManagedObject WHENEVER
        sourceManagedObject) ) then
        emit security/ServicOrMechanismViolation
        notification;
        abort ;
    endif;
REGISTERED AS {schedPkg 10};
    
```

(그림 6) dependencySchedulingPackage 패키지

```

ManagedObjectMode ATTRIBUTE GROUP
DESCRIPTION
    initiator GET-REPLACE,
    accesMode GET-REPLACE,
    target GET-REPLACE;
REGISTERD AS {smiAttributeGroup 3};
DependencyMode ATTRIBUTE
WITH ATTRIBUTE SYNTAX attribute
    ASNIModule.dependency;
REGISTERED AS {smiAttributeID 72};

dependency ::= CHOICE {UNLESS,
    WHENEVERNOT, ASLONGAS WHENEVER};
    
```

(그림 7) ManagedObjectMode 속성그룹과 DependencyMode 속성

• timestampSchedulingPackage

권고안에서 기술된 시간속성은 시작시간과 종료시간을 지정하여 수행하는 durationPackage나 dailySchedulingPackage 그리고 weeklySchedulingPackage 등이 있다. 하지만 이러한 패키지는 관리자에 의해 정적인 시간개념으로 부여가 되며 관리객체 또한 중요도에 따른 차별화를 두지 않아 정보의 보안성이 침해될 우려가 있다. 이러한 관리객체에 대해서는 접근시간을 실시간 환경에 적용될 수 있는 동적 개념으로 변환하므로서 보다 안전하게 유지할 필요가 있다. 여기에서는 관리자가 관리객체에 접근요청을 시작하는 시점을 기준으로 하여 종료시점을 부여하는 타임스탬프 개념을 도입하여 정보의 안전성을 도모하였다.

<정의 2> timestampScheduling 규칙

$(m_i, a_i, m_{o_i}, p, T_i)$
 단, $m_i \in M$: 관리자, $a_i \in A$: 대리자, $m_{o_i} \in MO$: 관리객체,
 $p \subseteq P$: ACCESS_MODE, $T_i \in \mathbf{N} \cup \infty$: EXPIRETIME

한 예로 다음 $(m_i, a_i, \{o_1\}, \{M-GET\}, 200)$ 규칙의 경우 관리자 m_i 이 대리자 a_i 의 관리객체 o_1 에 대해 접근요청을 시작한 이후부터 200초 동안에만 M-GET연산을 수행할 수 있다.

이를 근거로 한 timestampSchedulingPackage패키지와 expiredTime 속성은 다음과 같다.

```
timestampSchedulingPackage PACKAGE
BEHAVIOUR timestampSchedulingBehaviour
BEHAVIOUR
DEFINED AS
EVENT : checkedtimestamp
    timestampedInitiator : initiator,
    timestampedTarget : target,
    timestampedAccessMode : accessMode,
    timestampedExpiredTime : expiredTime;
PRECOND :
    administrativeState == unlocked AND
    operationalState == enabled AND
    availabilityStatus != Offduty;
PROCEDURE :
    if ( ( expiredTime != NULL ) AND ( OVER
        expiredTime ) ) then
```

```
emit
securityServiceOrMechanismViolation
notification;
abort ;
endif;
REGISTERD AS {schedPkg 11};
```

(그림 8) timestampPackage 패키지

```
expiredTime ATTRIBUTE
WITH ATTRIBUTE SYNTAX
attribute-ASNIModule.ExpiredTime;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR timeOrdering;
REGISTERD AS {smiAttributeID 73};

ExpiredTime ::= CHOICE { specific
                        GeneralizedTime,
                        continual NULL }
```

(그림 9) expiredTime 속성

3.2 관리객체의 시간지원 동적 특성 기술

여기에서는 ITU-T 권고안에서 정의하고 있는 관리자 관리 기능 모델에 대한 관리객체 클래스의 시간지원 동적특성을 본 논문에서 제안한 BDL을 이용하여 기술한 예를 살펴본다.

3.2.1 접근제어 기능

접근제어 기능은 관리객체의 보안을 위하여 사용자의 접근을 제어 및 관리하는 기능이다. 접근제어를 위한 메카니즘은 접근행렬, 접근제어 리스트, 능력리스트에 의한 방법인 자율적 접근제어와 관리자와 관리객체의 보안 등급에 의한 방법인 강제적 접근제어 그리고 관리자의 역할에 의한 역할기반 접근제어가 있다. rule 관리객체는 관리자가 특정 관리객체에 접근하고자 할 때 규칙에 근거하여 접근여부를 결정하는 기능을 수행한다. 현재 권고안에는 이러한 접근제어의 규칙 객체의 동작 과정을 자연어로 기술하고 있어 관리객체의 동작 조건이나 절차, 결과 등이 명확히 표현되지 못한 문제점을 가지고 있다. 이에 본 논문에서는 관리객체의 동적 특성을 기술하기 위한 프로그래밍언어형태의 정형적 기술

언어인 BDL을 이용하여 행위 템플릿을 기술하였다.

다음은 BDL를 이용한 접근제어 관리객체의 동적 특성중 규칙에 근거한 관리객체 접근을 제어하는 과정과 수행절차를 정형적으로 기술하고 있다.

```

rule MANAGED OBJECT CLASS
DERIVED FROM accessControl;
CHARACTERIZED BY rulePackage PACKAGE
BEHAVIOUR ruleBehaviour BEHAVIOUR
DEFINED AS
EVENT : AccessControlEnforcementEvent
    accessControlObject :
    accessControlObjectName;

INVARIANTS :
    startTime <= stopTime;
PRECOND:
    administrativeState == unlocked AND
    operationalState == enabled AND
    availabilityStatus != Offduty;

PROCEDURE:
if ( NOT ( EXISTS durationPackage AND
    currentTime VALID IN
    [startTime, stopTime] )
OR ( EXISTS dailySchedulingPackage AND
    currentDay VALID IN [startDay,
    stopDay] )
OR ( EXISTS weeklySchedulingPackage AND
    currentWeek VALID IN
    [startWeek, stopWeek])) then
    emit timeDomainViolation notification:
    abort:
if ( NOT( EXISTS externalScheduler ) ) then
    emit securityServicOrMechanismViolation
    notification:
    abort:
endif;
endif;
if (enforcementAction == allow) then
    "access is permitted";
    validAccessAttempts = validAccessAttempts + 1;
    "send to a security audit trail log";
    emit usageReport notification:
    if (enforcementAction ==
    deny with response) then
    "access is denied";
    
```

```

invalidAccessAttempts =
invalidAccessAttempts + 1;
"send to a security audit trail log";
emit usageReport notification:
endif:
endif:
endif:
    
```

(그림 10) BDL을 이용한 rule 관리 객체 클래스의 동작 특성 기술

(그림 10)에서 'rule' 관리객체 클래스는 'accessControlEnforcementEvent' 사건이 발생하면 먼저 'INVARIANTS'의 불변조건과 'PRECOND' 상태조건을 점검한 후 해당되는 동작절차를 수행한다. 동작절차에서는 해당 관리자의 관리객체에 대한 접근시간 항목을 점검한 후 시간속성이 맞지 않으면 시간영역위반의 통지를 발송하고 이 사건을 중지시킨다. 이 절차를 거친 후에 만약 'enforcementAction' 속성값이 'allow' 값을 갖으면 접근을 허용하고 보안 감사를 위해 'log' 관리객체에 'usageReport' 통지를 전송한다. 그렇지 않고 'enforcementAction' 속성값이 'deny with response' 값을 가지면 접근은 불허되고 이 또한 보안 감사를 위해 'log' 관리객체에 'usageReport' 통지를 전송한다.

3.2.2 사건 보고 기능

사건 보고 기능은 시스템에 이미 발생한 사건이나 발생할 가능성이 있는 사건 등을 관리자에게 통지해주는 역할을 수행한다. 사건에 대한 통지는 사건 전송 식별자(Event Forwarding Discriminator)에 의하여 선택되어 CMIS의 M-EVENT-REPORT 서비스를 이용하여 관리자에게 보내어진다[17]. 사건 보고 기능에서 발생하는 통지는 잠재적 사건 보고(Potential Event Report)로서 사건 전송 식별자의 제어를 통해 선택적으로 전송되는데 이때 전송되는 통지를 사건 보고(Event Report)라 한다. 사건전송 식별자는 관리객체 클래스로 식별자 구조(Discriminator Construct)와 목적지 주소(destination)의 두 가지 속성을 갖는다. 전자는 통지가 전송될 것인지를 결정하는 기능을 하며 후자는 선택된 사건 보고에 대한 목적지 주소를 갖고 있다. 이와 같이 선택적으로 사건보고에 대한 통지를 제공하는 기능과 함께 스케줄링을 통해 정해진 시간이나 일간, 주간에 대해 일정별로 보고를 처리해준다.

다음은 BDL를 이용한 사건 보고 기능에 대한 시간 지원 동적 특성을 정형적으로 기술한 예이다.

```

eventForwardDiscriminator MANAGED
                        OBJECT CLASS
DERIVED FROM discriminator;
CHARACTERIZED BY efdPackage PACKAGE
BEHAVIOUR eventForwardingDiscriminatorBehaviour
BEHAVIOUR
DEFINED AS
    EVENT : PortentialReportEvent
        managedObjectClass: ObjectClass,
        managedObjectInstance: ObjectInstance,
        eventType: EventTypeID,
        severity: SecurityAlarmSeverity,
        backedUpStatus: BackedUpStatus,
        probableCause: ProbableCause;

    PRECOND :
        administrativeState == unlocked AND
        operationalState == enabled AND
        availabilityStatus != offDuty;

    PROCEDURE :
        if ( (FILTER ( discriminatorConstruct,
                        pontentialEventReport ) ) AND
            (EXISTS durationPackage OR
             dailySchedulingPackage OR
             weeklySchedulingPackage) ) then
            emit eventReport notification;
        endif;

    EVENT : replaceDiscriminator
        inDiscriminatorConstruct: DiscriminatorConstruct;

    PRECOND:
        administrativeState == unlocked AND
        operationalState == enabled;

    PROCEDURE:
        discriminatorConstruct =
        inDiscriminatorConstruct;
        emit attributeValueChange notification;
    
```

(그림 11) BDL을 이용한 eventForwardDiscriminator MO Class 동작 특성 기술

(그림 11)에서 'eventForwardDiscriminator'관리객체

클래스는 'discriminator' 관리객체 클래스로부터 모든 속성을 상속받으며 'PotentialReportEvent' 사건이 발생하면 해당하는 상태조건을 먼저 점검한다. 상태조건이 합당하면 잠재적 사건보고와 식별자 구조를 필터링하는 과정과 시간속성값을 점검한 후 식별자의 정의에 부합된 잠재적 사건보고만이 사건보고 통지가 발생한다. 그리고 식별자 속성값을 변경하는 'replaceDiscriminator' 사건이 발생하면 입력으로 들어오는 식별자 구조를 새로운 식별자 구조로 할당하고 속성값의 변경을 알리는 'attributeValueChange' 통지를 발생한다.

3.2.3 스케줄링 기능

특별한 관리객체 인스턴스 내에 명세되어 있는 시간 관리객체의 활동을 제어하여 관리객체 내에서 자체적으로 발생하는 'trigger scheduling' 인 비주기적인 스케줄링과 관리객체의 활동에 사용되는 연산의 시간 구간을 지정하여 일정하게 반복적으로 제어하는 'interval scheduling' 주기적인 스케줄링으로 나뉜다. 특히 스케줄링을 수행하는 관리객체를 스케줄링 관리 객체(Scheduling Managed Object)라 하며 스케줄링 관리 객체에 적용되는 스케줄링 값과 타입을 관리하는 객체를 스케줄러 객체(Scheduler Object)라 한다. 스케줄링의 특성에는 내부적 스케줄링과 외부적 스케줄링으로 나뉘는데, 만약 관리객체 클래스가 이후에도 변경이 되지 않거나 스케줄링의 그룹핑이 가능하다면 스케줄러 객체 내부에 스케줄링 관리 객체의 스케줄링 기능을 구현하는 내부적 스케줄링으로 정의하는 것이 바람직하다. 그러나 클래스가 변경이 되거나 다양한 스케줄링 기능이 클래스에 포함된다면 각 기능별로 스케줄링을 구현하는 외부적 스케줄링 기능을 이용하는 것이 옳다. 일반적으로 하나의 스케줄러 객체가 수많은 스케줄링 관리 객체들의 활동을 제어하는 구조를 갖는다[20].

스케줄링 기능에 대한 스케줄러 관리객체의 시간 지원 동적 특성의 정형적 기술은 다음과 같다.

```

scheduler MANAGED OBJECT CLASS
DERIVED FROM : top;
CHARACTERIZED BY schedulerObjectPackage,
                    duration;
schedulerObjectPackage PACKAGE
BEHAVIOUR schedulerObjectBehaviour
    
```

BEHAVIOUR

DEFINED AS

EVENT : LockSchedulingManagedObject

schedulerObjectName : scheduledObject,
 schedulerManagedObjectName :
 scheduledManagedObjects,
 schedulerId : schedulerID;

PRECOND :

administrativeState == unlocked;

PROCEDURE :

```

if ( administrativeState CHANGES locked )
    then
        "interval scheduled action is terminated
        immediately";
        administrativeState = locked;
        emit stateChange notification;
    endif;
    
```

EVENT : ModifySchedulingManagedObject

schedulerObjectName : scheduledObject,
 schedulerManagedObjectName :
 scheduledManagedObjects,
 schedulerId : schedulerID,
 newStartTime : StartTime,
 newStopTime : StopTime,

PRECOND :

administrativeState == unlocked;

PROCEDURE :

```

if ( startTime CHANGES newStartTime )
    then
        startTime = newStartTime;
        emit stateChange notification;
    if ( stopTime CHANGES newStopTime )
        then
            stopTime = newStopTime;
            emit stateChange notification;
        endif;
    endif;
endif;
    
```

SchedulingManagedObject' 사건은 'administrativeState' 속성값이 'locked' 모드로 변경되면 수행중인 내부의 관리객체의 수행을 즉시 중단하고 'administrativeState' 속성값을 'locked'모드로 변경하고 'state Change' 통지를 발생한다. 'ShutdownSchedulingManaged Object' 사건은 'administrativeState' 속성값이 'shutdown' 모드로 변경되면 수행중인 내부의 관리객체의 수행은 일상적으로 수행하여 마친 후 'administrativeState' 속성값을 'locked'로 변경하고 'stateChange'통지를 발생한다. 그리고 'Modify SchedulingManaged Object' 사건은 시작시간과 종료 시간이 변경되면 새로운 시간으로 속성값을 변경한 후 'stateChange' 통지를 발생한다.

3.2.4 로깅 기능

다양한 객체에 의해 수행되어진 연산이나 이미 발생한 사건에 관한 정보를 유지할 필요가 있는데 OSI 관리 모델에서는 'log'나 'log record'라는 관리객체클래스를 사용하여 정보를 보관한다. 'log' 관리객체는 외부로부터 전달되는 사건 보고나 로컬 시스템에서 발생된 통지의 저장 기능을 수행한다. 'log record'는 로그에 저장되는 정보의 단위를 나타낸다. 로그 관리 기능(Log Control Function)에 대한 X.721 권고안에 기술된 'log' 관리객체의 동적 특성을 본 논문에서 제안된 BDL을 이용하여 기술한 예를 살펴본다. 'log' 관리객체는 외부로부터 전달되는 사건 보고나 로컬 시스템에서 발생된 통지의 저장 기능을 수행한다. 그러나, 관리객체의 동작상태를 나타내는 'administrativeState', 'operationalState', 'availabilityStatus' 속성값이나 'discriminator Construct' 속성값에 의해 관리객체로 전달된 통지나 사건 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.

다음은 BDL을 이용한 로그 관리객체의 동적 특성 중 다른 객체로부터 전달된 통지에 대한 처리과정과 통신망 관리자로부터 전달된 속성값 변경에 대한 관리 명령 수행 절차를 나타내고 있다.

(그림 12) BDL을 이용한 scheduler MO Class의 동작 특성 기술

스케줄러 관리 객체는 'LockSchedulingManaged Object' 사건, 'ShutdownSchedulingManagedObject' 사건, 그리고 'ModifySchedulingManagedObject' 사건으로 나뉜다. 'Lock

```

logPackage PACKAGE
BEHAVIOUR
logBehaviour BEHAVIOUR
DEFINED AS
EVENT: internalNotification
      notification: smi2Notification,
      notificationID: NotificationIdentifier;
PRECOND:
      administrativeState == unlocked AND
      operationalState == enabled AND
      availabilityStatus != offDuty AND
      availabilityStatus != logFull;
PROCEDURE:
      logRecordSize : INTEGER;
      "set the logRecordSize according to data
      logged";
      if ( NOT ( FILTER ( discriminatorConstruct,
                        notification ) ) ) abort;
      endif;
if ( NOT ( EXISTS durationPackage AND
      currentTime VALID IN [ startTime, endTime ] ) )
then
      abort;
endif;
if ( EXISTS finiteLogSizePackage ) then
      "process the notification suitable to log record
      structure, add modified notification information to
      log storage";
      numberOfRecords = numberOfRecords + 1;
      currentLogSize = currentLogSize + logRecordSize;
      if ( currentLogSize + logRecordSize ==
            maxLogSize ) then
            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
      inDiscriminatorConstruct: DiscriminatorConstruct;
PRECOND:
      administrativeState == unlocked AND
      operationalState == enabled;
PROCEDURE:

```

```

discriminatorConstruct = inDiscriminatorConstruct;
emit attributeValueChange notification;
...

```

(그림 13) BDL을 이용한 log 관리객체 클래스의 동작 특성 기술

(그림 13)에서 '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 관리객체에 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.

4. 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.

표준통신망 구성요소를 기술하는 현재의 표준안이 관리객체의 구조나 속성 등 정적인 부분은 정형적으로 기술되어 있지만 관리객체에 동작에 영향을 미치는 능동적 동작절차를 표현하는 행위 부분이 자연어를 이용하여 매우 포괄적이며 간략히 기술되어 있어, 관리객체의 동작 조건이나 절차, 결과 등이 명확히 표현되지 못하는 문제점을 가지고 있다. 또한, 관리객체의 동적 특성은 능동적 특성 외에 시간지원 특성에 따라 동작절차가 달라질 수 있다. 관리객체의 시간지원 동작절차는 통신망 관리 목적이나 관리객체의 특성에 따라 정적인 시간개념인 지정된 기간 또는 반복적인 작동주기를 가지고 동작될 수 있다. 이러한 정적인 시간개념 뿐만 아니라 다른 관리자의 속성에 따라 사용권한이 결정되거나 특정기간 동안에만 권한이 부여되는 동적인 시간개념이 첨가되어 관리객체의 능동적 특성을 더욱 강화시킬 필요가 있다.

본 논문에서는 정적인 시간 관리객체의 활동을 제어하는 표준권고안의 스케줄링 관리객체에 동적인 시간지원 특성을 추가하여 관리객체 동적특성을 더욱 강화시켰었으며 관리객체 기술의 표준인 GDMO 표현법에 의해 동적특성을 지원하는 스케줄링 관리객체를 기술하였다. 또한 관리객체의 동적 특성의 기술을 위해 새롭게 정의한, 동적 시간지원 기능이 추가된 BDL을 이용하여 동작절차를 기술하였다. 그리고 동적 시간지원 특성을 지원하는 관리객체를 대리자 관리 기능 모

델에 적용하여 봄으로서 관리객체의 동적 수행 기능이 보다 체계적이고 정형적으로 기술되는 장점을 보였다. 제시된 정형적 기술방법은 통신망 관리 프로그램의 자동생성과 시험에 활용될 수 있을 것이며, 제안된 정형적 기술언어의 구조와 기능을 보완하고, 기술언어 각 구성요소의 의미에 대한 정형화 연구와 함께 정형화 검증을 진행할 계획이다.

부 록

동적 시간지원 기능이 추가된 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
Temporal-Exp ::= Attribute Temporal-Op Temporal-Interval
              | 'currentTime' Temporal-Op Temporal-Interval
              | Attribute Temporal-Op Attribute
              | Temporal-Op Attribute
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'
             | 'UNLESS'
             | 'WHENEVERNOT'
             | 'ASLONGAS'
             | 'WHENEVER'
             | 'OVER'

Type-Name ::= { x | x is attribute type in
Attribute-ASNIModule }
Exp ::= Number
| Attribute

```

```

| (' Primary-Exp ')

Primitive-Set-Exp ::= { x | x is ANY type }
| Attribute
| (' Primary-Set-Exp ')

Primary-String-Exp
Attribute-ASNIModule.AdditionalText
| Attrib

| (' String-Exp ')
Number ::= INTEGER
        | REAL

Time-Point ::= Attribute-ASNIModule.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-ASNIModule.AdditionalText ""

```

참 고 문 헌

- [1] Elisa Bertino, Claudio Bettini, Pierangela Samarati, "A Discretionary Access Control Model with Temporal Authorizations," IEEE New Security Paradigms Workshop, Aug., 1994.
- [2] Elisa Bertino, Claudio Bettini, Pierangela Samarati, "A Temporal Authorization Model," 2nd ACM Conference on Computer and Communications Security, Nov., 1994.
- [3] Elisa Bertino, Claudio Bettini, Elena Ferrari and Pierangela Samarati, "An Access Control Model Supporting Periodicity Constraints and Temporal Reasoning," ACM Transaction on Database Systems, Vol.23, No.3, Sep., 1998.
- [4] 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.
- [5] Oliver Festor, Georg Zornlein, "Formal Descri-

- ption of Managed Object Behavior-A Rule Based Approach,” Proceedings of the IFIP TC6/WG6.6 3rd International Symposium on Integrated Network Management, 1993.
- [6] Masum X. Hasan, “An Active Temporal Model for Network Management Databases,” Proceedings of the 4th International Symposium on Integrated Network Management, 1995.
- [7] ISO International Standard 7498 Information Processing Systems-Open Systems Interconnection-Basic Reference Model Part 4: Management Framework 1991.
- [8] Christian S. Jensen, Richard T. Snodgrass, “Temporal Data Management,” IEEE Transactions on knowledge and data engineering, Vol.11, No.1, Jan., 1999.
- [9] J. Keller, “An Extension of GDMO for Formalizing Managed Objects Behaviour,” Proceedings of the IFIP TC6 8th International Conference on Formal Description Techniques, 1995.
- [10] Dennis R. McCarthy, Umeshwar Dayal, “The Architecture Of an Active Data Base Management System,” ACM SIGMOD International Conference on the Management of Data, Vol.18, No.2, 6, 1989.
- [11] Iakovos Motakis, Carlo Zaniolo, “Composite Temporal Events in Active Database Rules : A Logic-Oriented Approach,” Fourth International Conference DOOD, Dec., 1995.
- [12] James Rumbaugh, Michael Blaha, William Premerlani, Frederick Eddy, William Lorensen, Object-Oriented Modeling and Design, Prentice-Hall, 1991.
- [13] Arie Shoshani, Kyoji Kawagoe, “Temporal Data Management,” Twelfth International Conference On Very Large Data Bases, Aug., 1986.
- [14] Morris Sloman, Network and Distributed Systems Management, Addison-Wesley, 1994.
- [15] Adrian. Tang et al. Open Networking with OSI, Prentice-Hall, 1992.
- [16] Divakara K. Udupa, Network Management Systems Essentials, McGraw-Hill, 1996.
- [17] ITU-T X.720|ISO DIS 10165-1 : “Information Processing Systems-Open Systems Interconnection-Management Information Model,” Geneva
- [18] 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.
- [19] 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.
- [20] 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.”
- [21] ITU-T X.725 : “Information Processing Systems-Open Systems Interconnection-Structure of Information : General Relationship Model.”
- [22] ITU-T X.734|ISO DIS 10164-5 : “Information Technology-Open Systems Interconnection-System Management-Event Report Management Function.”
- [23] ITU-T X.735|ISO DIS 10164-6 : “Information Technology-Open Systems Interconnection-System Management-Log Control Function.”
- [24] ITU-T X.741|ISO DIS 10164-9 : “information Technology-Open Systems Interconnection-System Management-Objects and Attributes for Access Control.”
- [25] ITU-T X.746|ISO DIS 10164-15 : “information Technology-Open Systems Interconnection-System Management-Scheduling Function.”
- [26] 최은복, 이형효, 노봉남, “망관리 객체의 시간자원 능동특성에 대한 정형적 모델링”, 정보처리학회 논문지 제6권 제9호.



최 은 북

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

관심분야 : 통신망관리, 정보보안, 멀티미디어시스템 등



이 형 호

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)
 2000년 전남대학교 전산학과 졸업(이학박사)
 관심분야 : 통신망관리, 정보보안, 객체지향시스템 등



노 봉 남

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, 컴퓨터
 와 정보사회 등