

스크립트 관리 기능의 설계 및 구현에 관한 연구

정회원 한 순 희*, 정회원 이 재 오** 정회원 조 국 현**

A Study on Design and Implementation of Script Management Function

Soon Hee Han,* Jae Oh Lee,** Kuk Hyun Cho** *Regular Members*

要 約

위임 모델은 OSI 네트워크 관리 모델에서의 관리 정보 전송으로 인한 과도한 정보 전송을 줄임으로서 효율적인 네트워크 관리를 지원하고, 신뢰도를 높이기 위한 모델이다. 이 모델에서는 관리 스크립트를 작성하여 이의 수행을 피관리자에게 위임함으로써 관리자의 부담을 줄이려고 노력하였다. 따라서 스크립트의 효율적인 작성과 위임등 여러가지 기능이 필요하게 된다. 본 논문에서는 효율적인 네트워크 관리를 위한 관리 스크립트 언어를 설계하고, 인터프리터를 구현한다. 그리고 스크립트 관리를 위한 기능 및 이를 실현하기 위한 서비스를 정의하고, 관리자와 피관리자 모델과 알고리즘을 제시하며 ISODE를 이용한 실험적인 구현을 통하여 그 기능을 확인한다.

ABSTRACT

Delegation model supports effective and highly reliable network management. It reduces traffic overhead caused by transmission of management informations. In this model, management scripts can be composed by the designers of managers, these are delegated to agents. Delegation models support effective distribution of management functions among managers and agents. Therefore, various functions are needed to create and delegate management scripts.

In this paper, we design simple script language and implement script interpreter for efficient network management. And, we define script management functions and services. Moreover, we present manager /agent models and algorithms for script management. As an implementation environment, the ISODE services are ported. The methods employed for the implementation are also described.

* 麗水水産大學 컴퓨터工學科
Dept. of Computer Engineering, Yosu National Fisheries
University
** 光云大學校 電子計算學科
Dept. of Computer Science, Kwangwoon University
論文番號 : 93-75

I. 서 론

네트워크의 규모가 커지고 사용이 증가되면서 네트워크 관리의 필요성이 대두되었고, 이에 따라 각

제조업체에서는 자사의 네트워크 장비에 적합한 관리 시스템을 구축하였으나, 이기종 시스템으로 구성된 경우에는 관리에 많은 어려움을 겪게 되었다. Y. Lamdan 등은 각 제조업체가 제공하는 네트워크 관리 시스템을 통합하여 관리하기 위한 인터페이스를 정의하고 새로운 관리 언어를 설계하기도 하였다.⁽¹⁾ 한편 이러한 어려움을 타개하기 위하여 표준화 작업이 진행되어 왔으며, ISO /IEC에서는 OSI 참조 모델을 근거로 한 OSI네트워크 관리 모델을 제시하였다.^(2,3) 그러나 표준화를 위한 대부분의 노력은 관리 프로토콜과 관리 정보의 구조에 집중되었으며, 관리자와 피관리자 간의 소프트웨어 구조에 대해서는 명시적으로 정의하지 않았다. 표준화 모델에서 관리자는 관리 프로토콜을 통하여 피관리자와 관리 정보를 교환하고, 이는 micro-management의 문제점을 야기시키게 되며, 관리 정보의 교환으로 인하여 트래픽 오버헤드가 따르게 된다.⁽⁴⁾ 이러한 오버헤드를 줄이기 위한 많은 연구가 진행되어 왔으며, 그 중의 한 방법으로 Yemini 등에 의하여 위임 모델이 제시되었다.^(4,5)

위임 모델에서 관리자는 필요한 경우에 피관리자에게 스크립트의 수행을 위임할 수 있으며, 이러한 위임을 통하여 관리자와 피관리자 간의 트래픽의 현저한 감소 뿐만이 아니라 관리자측에서 관리 스크립트를 작성하는 것을 가능하게 하였다. 그러나 Yemini 등이 제시한 방법에서는 위임만을 지원할 뿐 스크립트 관리에 필요한 여러가지 사항들은 배제하였다. 따라서 본 논문에서는 위임 뿐만이 아니라 스크립트 관리를 효율적으로 수행하기 위한 스크립트 관리 기능을 설계하고, 이에 필요한 서비스에 제공하고 자 한다. 본 논문은 II장에서 OSI 네트워크 관리 모델과 위임 모델에 대해 기술하고, III장에서는 스크립트를 관리하기 위한 스크립트 관리 기능의 설계에 대해 논하고, 스크립트 관리를 위한 관리자와 피관리자 모델 및 알고리즘을 제안하며, 설계한 서비스의 구현에 대해 각 모듈별로 간략히 설명한다. 그리고 IV장에서는 구현 예를 보이고 결과를 고찰하며, 마지막으로 V장에서 결론을 맺는다.

II. 위임 모델

2.1 네트워크 관리를 위한 OSI모델

응용계층은 OSI 참조 모델의 최상위 계층으로서 응용처리를 하는 응용 프로세스 간의 인터페이스나

통신을 위한 기본적인 응용 기능을 제공하고, 응용 프로그램에 따라 여러가지 서비스를 제공한다. 이를 응용 서비스 요소라 하며, 응용 서비스 요소에 따라 서비스가 정의되고, 서비스를 실현하기 위한 응용 프로토콜 및 표현층(presentation layer) 서비스의 사용 방법이 규정된다. 응용 서비스 요소는 다른 응용 서비스를 호출하여 이용하는 서비스도 있으며, 다른 응용 서비스가 공통으로 사용하는 공통 응용 서비스(CASE:Common Application Service Element)와 특정 기능에 대응한 특정 응용 서비스(SASE: Specific Application Service Element)의 두가지가 있다. 특정 응용 서비스 중 OSI 네트워크 관리는 개방 시스템 간의 통신을 효율적으로 운영하고, 네트워크 내의 자원을 감시하여, 비정상적인 상태를 제어함으로써 오류없이 실현하고자 하는 기능이며, ISO /IEC에서는 OSI 관리 구조(Management Framework)를 제시하였다. OSI 관리는 시스템 관리, N층 관리, N층 조작에 의해 제공되는 기능으로 수행되고, OSI 관리 구조에서 시스템 관리는 관리 프로세서 간에 교환될 응용층 시스템 관리 프로토콜을 이용하여 관리 객체를 감시하거나 통제하는 등의 메커니즘이며, 개방 시스템 내의 관리 객체들과 관련된 관리 정보를 저장하기 위하여 MIB(Management Information Base)를 이용한다.^(2,3,6,7) 이 권고안에서는 관리 구조는 네트워크 관리를 위해 5개의 기능 영역으로 분리하고, 이를 장애 관리(Fault Management), 구성 관리(Configuration Management), 성능 관리(Performance Management), 계정 관리(Accounting Management), 보안 관리(Security Management)라 칭하였다. OSI 관리 모델은 관리자와 피관리자가 관리를 위하여 필요한 정보를 주고 받게 되며, CMIS(Common Management Information Service)는 관리 association 제어 기능, 관리 객체의 생성과 삭제, 사건 보고, 관리정보 전송기능, 관리 객체 제어 기능 등을 제공한다.⁽⁷⁾ 특정 응용 관리 서비스(SMIS: Specific Management Information Service)는 이러한 기능들을 사용하여 네트워크 내의 관리 객체를 관리하고 제어하게 된다.

2.2 위임 모델

컴퓨터 네트워크의 규모가 방대해지고 복잡해짐에 따라 오류나 시스템의 성능과 관련된 비효율성은 기업의 부담을 과중시키는 요인이 되고 있다. 따라서 시스템을 오류없이 효율적이고 안전하게 유지하는

것이 큰 과제로 부각되고 있으며, 제조업자, 표준위원회, 연구소 등에서 이를 위한 많은 노력을 기울이고 있다. 그러나 많은 컴퓨터 네트워크 시스템은 관리하기에 적절한 형태로 구성되어 있는 것은 아니며, 네트워크 관리에 많은 어려움을 주고 있다. 2.1에서 설명한 바와 같이 통상적인 네트워크 관리 시스템은 관리 객체들을 감시하고 통제하는 피관리자(agents)와 피관리 시스템으로부터의 동적으로 발생하는 자료를 수집하고 자료를 분석하여 적절한 행위를 취할 것을 피관리 시스템에 지시하는 관리자(managers) 시스템으로 구성된다. 적절한 관리를 수행하기 위해서 관리자와 피관리자는 관리 프로토콜(management protocol)을 통해서 상호 동작하며, 관리 객체에 적용할 감시나 통제, 동작 등을 기술하기 위해서는 관리 기술 언어(management scripting language)를 사용한다. 관리 기술 언어를 사용하여 관리 행위를 기술한 것을 관리 스크립트(management script)라 한다.⁽⁸⁾

OSI CMIS 관리 모델에서는 관리자와 피관리자간의 interaction 프리미티브를 통해서 관리를 수행하게 된다. 관리 객체 인스턴스의 생성, 삭제, 검색을 위해서는 M-CREATE, M-DELETE, M-GET을 이용하고, 관리 객체 속성의 지정과 사건 보고를 위해서는 M-SET, M-EVENT-REPORT를 관리 행위를 시작하기 위해서는 M-ACTION을 각각 이용하게 된다. OSI 네트워크 관리 모델은 관리 프로토콜과 이를 지원하기 위한 관리 정보 모델(SMI : Structure of Management Information model)에 대해서만 표준화 노력을 기울였으며, 이를 지원하기 위한 관리자와 피관리자의 소프트웨어 구조에 대하여는 명시하지 않았다. 따라서 관리자와 피관리자 간에 관리 책임을 어떤 방법으로 분담할 것인가? 관리자와 피관리자 간에 관리 행위를 어떤식으로 상호동작 시킬 것인가? 피관리자는 관리 객체의 어떤 행위를 감시하고 통제할 것인가? 등의 의문점이 남게 된다. 이러한 문제들은 효율적인 관리를 위해서는 필연적으로 해결되어야 할 문제점들이다.

표준화 모델에서는 모든 관리 기능이 관리자에게만 집중되게 되므로, 사소한 장애도 관리자에게는 과중한 부담이 되며 특히 관리자에게 장애가 발생했을 시는 피관리자는 관리자로부터의 관리 명령을 기다려야 함으로 회복할 수 없게 된다. 이와 같은 집중화 현상은 대규모고속 네트워크 시스템의 관리를 불가능하게 하는 요인이 될 수 있다.

이러한 문제점을 해결하기 위한 방안으로 전체 스크립트를 피관리자에게 전송하고 이를 피관리자가 수행하는 방법이 있다. 이를 위한 관리자와 피관리자의 역할에 대해 살펴보면 우선 전형적인 방법으로 원격 프로시저 호출(remote procedure call)을 고려할 수 있다. 관리 스크립트가 피관리자 코드로서 encode 되어 있을 때 관리자는 원격 프로시저 호출을 통하여 피관리자로 하여금 관리 스크립트의 수행을 시작하게 할 수 있다. 그러나 원격 프로시저 호출시 관리자는 블럭되게 되고 피관리자의 수행이 끝났을 때 재개된다. 또 관리 스크립트가 피관리자의 입장에서 설계된 것이므로, 피관리자 시스템 설계자는 설계시 피관리자 프로시저를 작성해야 하는 부담을 가지게 되며, 피관리자 설계시에 이러한 모든 상황을 예측한다는 것은 불가능한 일이다. 더구나 피관리자 시스템 내에 이러한 코드를 작성하여 두는 것은 비용의 낭비를 초래하며 피관리자 시스템의 복잡성을 증가시키게 된다. 두번째로 고려할 수 있는 방법은 원격 수행(remote evaluation)인데 관리자가 관리 스크립트를 작성하여 전송하는 방법이다. 이는 관리 스크립트를 피관리자 관점에서 작성해야 하는 부담은 제거될 수 있으나 원격 프로시저 호출에서와 마찬가지로 동기화의 문제는 그대로 남게되므로 피관리자가 수행을 끝낼 때까지 관리자는 블럭되어야 한다. 반면에 위임 모델은 전체 관리 스크립트를 피관리자에게 위임하고 관리자와 관계없이 피관리자가 관리 스크립트를 수행하는 모델이다. 따라서 피관리자는 사건 발생 감지부터 지역 행위(local action)의 수행에 이르기까지 모든 것을 책임지게 되므로 관리자와 피관리자 간에 관리 책임을 적절하게 분산하게 하는 기능을 지니게 된다. 여기에서 관리 기술 언어로서 표현될 수 있는 관리 스크립트는 관리자에 의해 설계되며, 관리자는 필요에 따라 피관리자에게 수행을 위임할 수 있다. 일단 위임된 관리 스크립트는 관리자의 방해없이 피관리자에 의해 수행되며, 관리자는 위임 프로토콜을 사용하여 관리 프로그램을 피관리자에게 위임하고, 위임된 관리 프로그램은 피관리자의 환경하에서 수행되게 된다. 관리 객체에 대한 액세스는 피관리자의 통제하에서 수행되며, 관리자와 피관리자 간에서는 수행의 중지나 재개 등을 위한 기법이 유지되어야 한다.

III. 스크립트 관리 기능의 설계 및 구현

3.1 스크립트 관리 기능의 설계

컴퓨터 네트워크를 관리하기 위해서는 효율적인 관리 스크립트의 작성이 필요하고, 이에 대한 관리가 요구되어 진다. 특히 2장에서 설명한 바와 같이 관리자 관점에서 관리 스크립트를 작성하여 피관리자 시스템에게 위임하고자 할 경우 작성된 관리 스크립트에 대한 것 뿐만이 아니라 위임에 관한 모든 정보를 유지하고 있어야 하며, 위임된 스크립트의 수정이나 위임 취소 등을 지원하기 위한 수단이 필요하게 된다.

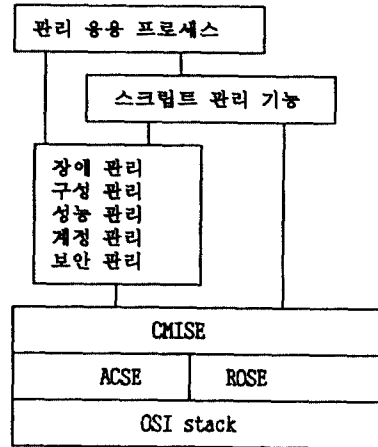
이러한 요구에 따라 본 논문에서는 스크립트 관리를 위해 필요한 기능을 정의하고 이러한 기능을 지원하기 위한 스크립트 관리 모델 및 서비스를 결정하고 설계하고자 한다.

3.1.1 스크립트 관리 기능 및 구조

스크립트 관리 기능은 관리자가 필요시 동적으로 관리 스크립트를 생성할 수 있으며, 관리자가 다른 관리 기능을 수행하기 위해 특정한 관리 기능을 수행하기 위한 여유가 없을 경우나, 과도한 관리 트래픽으로 인한 시스템의 부담을 줄이고자 할 경우에 관리 스크립트의 수행을 피관리자 시스템에게 위임할 수 있으며, 긴급한 상황의 발생으로 인해 위임받은 피관리자 시스템의 수행을 일단 정지시키거나 재개할 수 있는 등의 역할을 한다. 또 위임받은 스크립트의 일부를 수정하는 일도 행하게 된다. 이러한 기능들을 수행하기 위한 기능 단위를 본 논문에서는 앞으로 스크립트 관리 기능(script management function)이라 부르기로 한다. 위임 모델은 OSI 관리 모델의 확장된 개념이므로 스크립트 관리 기능을 추가한 위임 관리 모델의 구조(architecture)를 그림 1과 같이 설계하였다.

그림 1에서 관리 응용 프로세스가 스크립트 관리 기능을 이용할 경우에는 스크립트 관리 기능을 호출하게 되고, 사용자가 특정 관리 기능(Specific management function)을 수행하고자 할 경우 연결 제어 서비스 요소(ACSE)를 이용하여 연결을 확립한 후 공통 관리 정보 서비스(CMISE) 요소에서 제공하는 프리미티브를 이용하여 관리 목적을 달성하게 된다. 연결 제어 서비스는 CMSIE 서비스 이용자로 하여금 연결 제어를 할 수 있는 기능을 제공하며, 응용 엔티티간의 연결 설립, 해제 등을 위한 A-ASSOCIATE, A-RELEASE, A-ABORT, A-P-ABORT 등의 프리미티브를 제공한다. 그리고 원격 동작을 수행하기 위

해서는 원격 동작 서비스 요소(ROSE)에서 제공하는 RO-INVOKE, RO-RESULT, RO-ERROR, RO-REJECT-U, RO-REJECT-P 등의 서비스 프리미티브를 사용할 수 있다.



CMISE : Common Management Information Service Element
 ACSE : Association Control Service Element
 ROSE : Remote Operation Service Element

그림 1. 스크립트 관리를 지원하기 위한 확장된 관리 모델
 Fig. 1. Extended management model supporting script management

3.1.2 스크립트 관리 서비스

스크립트 관리를 위해 제공하는 서비스는 관리자가 작성한 스크립트를 관리하기 위한 것으로서 스크립트 수행의 위임통지, 위임해지, 관리 스크립트의 수정과 수정통지, 위임된 스크립트 수행의 중지와 재개 등으로 이루어진다. 이들 서비스를 간략하게 설명하면 다음과 같다.

1. 위임(delegate) : 관리자가 작성한 관리 스크립트를 피관리자 시스템이 관리자와의 통신을 행하지 않고 독립적으로 시행할 수 있도록 지원하기 위하여 스크립트 수행권리를 피관리자 시스템에 위임하는 서비스이다.
2. 위임통지(delegation report) : 피관리자가 관리자로부터 스크립트 수행을 위임받았음을 관리자에게 통지하는 서비스이다.

3. 위임해지(abandon) : 위임된 스크립트 수행을 중지시키는 것으로, 긴급한 관리 상황의 발생이나 다른 스크립트 수행을 위임하기 위하여 이용할 수 있다.
4. 수정(change) : 위임된 관리 스크립트의 내용이나 우선 순위 등을 변경하고자 할 때 이용할 수 있다.
5. 수정통지(attribute change report) : 스크립트의 속성이 변경되었음을 관리자에게 통지하는 서비스이다.
6. 중지/재개(suspend/resume) : 위임된 관리 스크립트의 수행을 중지시키거나 중지된 관리 스크립트 수행을 재개하기 위한 기능을 제공하는 서비스이다.

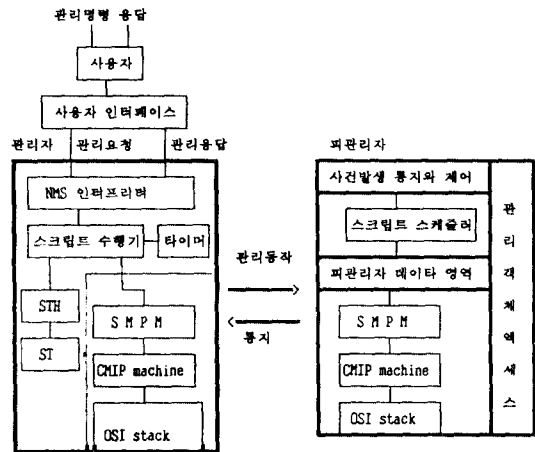
위의 설명한 각 서비스에 이용할 파라미터를 다음과 같이 설정하였으며, 각 파라미터를 살펴보면 다음과 같다.

- script-id : 네트워크 관리를 위해 작성된 스크립트명을 나타낸다.
- delegate-time : 위임된 시간을 표시하는 것으로 동기화를 위한 정보로 이용된다.
- delegator-id : 스크립트의 수행을 피관리자에게 위임한 관리자명을 나타낸다.
- delegate-count : 동일한 스크립트를 하나 이상의 피관리자에게 위임하게 되는 경우를 표시하기 위한 것으로 위임이 모두 해지되었을 경우에만 스크립트의 삭제 가능하다.
- severity-level : 스크립트 수행의 긴급도를 나타내기 위한 것으로 피관리자가 이미 다른 스크립트의 수행을 위임받아 수행 중 또 다른 스크립트 수행을 요청받았을 때 이 정보를 이용하여 수행할 스크립트를 결정한다.

3.2 스크립트 관리를 위한 관리자와 피관리자 모델

스크립트 관리를 지원하기 위한 관리자와 피관리자 관리 객체 간의 관계는 그림 2와 같으며, 이 모델에서 관리자와 피관리자는 스크립트 관리 요구에 따라 관리 기능을 수행하게 된다. 첫째 스크립트 관리를 위한 관리자 모델에서 관리자는 사용자의 요청에 따라 관리에 필요한 스크립트를 작성하게 되고, 작성

된 관리 스크립트는 검색 과정을 거쳐 스크립트 표(script table)에 기록된다. 한편 관리자가 스크립트의 위임이 필요하면 CMIS 서비스를 이용하여 상대편 피관리자와 통신을 하게 된다.



ST : Script Table
 STH : Script Table Handler
 SMPM : Specific Management Protocol Machine
 NMS 인터프리터 : Network Management Script 인터프리터

그림 2. 스크립트 관리를 위한 관리자/피관리자 모델
 Fig. 2. Manager/Agent model for script management

그림 2에서 NMS 인터프리터는 스크립트 관리를 이용하고자 하는 사용자에게 명령어(command)와의 인터페이스를 지원하기 위한 것으로서 명령어를 파싱하고 문법적인 오류를 검사한다. 오류가 있는 명령어는 사용자에게 오류 메시지를 보내며, 오류가 없을 경우 명령어를 스크립트 수행기(script executor)로 전달한다. 스크립트 수행기는 스크립트 관리 시스템 내의 정보 흐름을 통제하는 기능을 수행하며, NMS 인터프리터로부터 명령어를 받고, 시간에 관한 정보를 읽어 동기화에 관한 일을 수행하게 된다. 스크립트 표 handler(STH : script table handler)는 작성한 관리 스크립트에 관한 모든 정보를 저장하거나 검색해 주는 역할을 수행하기 위한 모듈로 관리 스크립트의 등록, 삭제, 갱신, 검색, 위임 등의 스크립트와 관련된 표를 유지하는 기능을 수행한다. 앞에서 설명한 스크립트 관리 기능을 수행하기 위한 관리자 알고리즘을 흐름도로 표시하면 그림 3과 같다. 흐름도에서 사각형의 윗부분에 기술된 내용은 모델에서의 각 모듈명을 나타낸다.

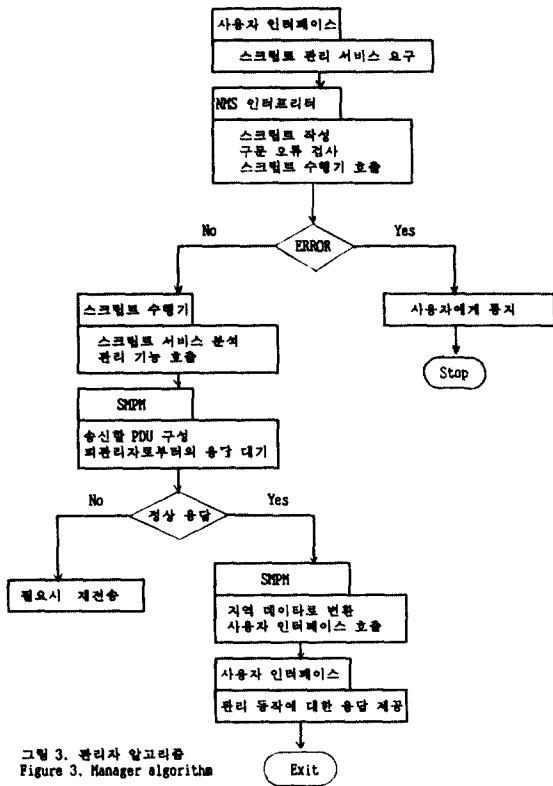


그림 3. 관리자 알고리즘
Figure 3. Manager algorithm

그림 3. 관리자 알고리즘
Fig. 3. Manager algorithm

그림 2에서 피관리자는 스크립트 관리를 위하여 관리자와 통신을 하게 되며, 관리자가 스크립트 수행을 위임하면 위임 표에 스크립트와 관련된 상황을 기록하고, 객체 관리를 위해 필요한 동작을 취하게 된다. 스크립트 스케줄러는 위임이나 위임해지 또는 중지/재개의 요청을 받았을 때 피관리자가 수행하는 일로서, 먼저 위임일 경우에는 위임된 스크립트와 관련된 모든 정보를 위임 표에 기록한 후 스크립트를 수행하기 위한 프로세스를 생성한다. 여러 관리자로 부터 하나 이상의 관리 스크립트의 위임을 받았을 경우에는 우선순위를 참조하여, 우선 순위가 가장 높은 스크립트를 수행하며, 이미 다른 스크립트 프로그램을 수행 중 더 높은 우선 순위의 스크립트 수행 요청이 있으면 수행 중인 스크립트 수행을 중지하고, 이를 위임자에게 통지한 후 새로운 스크립트를 수행한다. 관리 객체 액세스 부분은 수행이 결정된 관리 스크립트를 실제의 관리 객체에 관해 동작할 수 있도록 해

주는 역할을 담당하는 모듈로 피관리 시스템의 프로시저를 호출한다. 스크립트 관리 기능을 수행하기 위한 피관리자 알고리즘을 흐름도로 나타내면 그림 4와 같다.

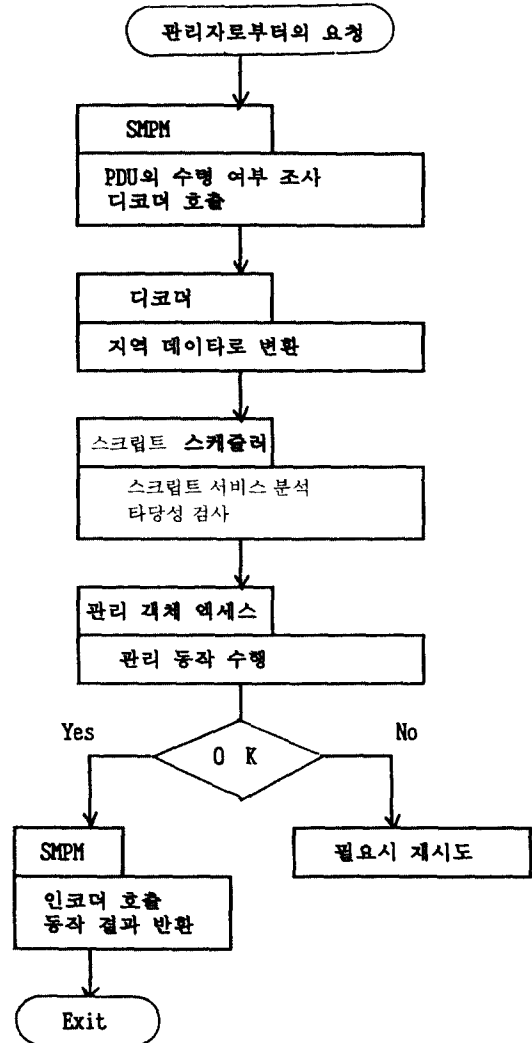


그림 4. 피관리자 알고리즘
Fig. 4. Agent algorithm

3.3 구현

본 절에서는 앞 절에서 설계한 모델에 따라 스크립트 관리 기능의 구현에 관해 각 구성 요소별로 이들

의 기능 및 구현과정을 간략히 설명한다. 구현 환경으로는 ISODE가 이식된 SUN4 시스템을 사용하였으며, 관리 정보 베이스로는 관계 데이터 베이스 형태의 자료를 이용하였고, NMS 인터프리터 구현은 Unix 시스템에서 제공하는 lex와 yacc를 이용한다. 그리고 스크립트 수행기와 타이머와의 통신을 위해서는 IPC를 사용하였다. 각 모듈별로 그 기능 및 구현 사항을 설명하면 다음과 같다.

3.3.1 공통 관리 정보 프로토콜 기계와 경보 보고 서비스

스크립트 관리 기능을 지원하기 위한 응용층 서비스 요소들로서, SMFA 중 장애 관리기능을 이용하며, ISODE의 네트워크 구조에 근거하여 OSI 관리 모델에 따라 구현하였으며, 응용층 구조인 ACSE, ROSE, CMISE, ALARM Reporting의 응용 엔티티로써 구성되어 있다. 관리자와 피관리자 사이의 association을 설정하기 위해서는 ISODE 라이브러리에서 제공하는 함수들을 이용하였고, 공통관리 정보 서비스를 구현하기 위해서는 ISODE에서 제공하는 ROSY와 POSY, PEPY 등을 이용하여 엔코딩/디코딩 루틴을 생성하였다. 경보 보고 서비스를 지원하기 위한 ALARM Reporting 엔티티는 하나의 경보 수령 프로세서인 alarmd와 여러개의 경보 보고 프로세서인 alarm으로 나누어 구성된다. 여러가지 장애 상황들에 대한 발견, 복구, 통지 등의 역할을 수행하기 위한 기능들은 스크립트로 작성하여 위임하거나, 관리자측에서 지역적으로 수행되도록 하였다. 스크립트 관리를 위해 각 서비스별로 입출력 매개변수와 C structure를 정의하였고, 사용자 인터페이스를 이용하여 위임, 위임해지, 중지 등의 operation을 요구하며, 경보 보고 서비스를 이용하여 사건 통지를 받는다. 요구된 operation은 원격 동작 서비스를 이용하여 서비스 공급자의 동작 수행 모듈을 호출하여 원하는 일을 수행하게 되는데, PDU로 정의된 operation모듈을 ROSY로 입력하고, 생성된 .py 파일을 입력받아 POSY에서는 추상구문과 일치하는 C structure로 바꾸어 준 후 PEPY를 이용하여 POSY에 의해 생성된 파일을 원격 동작에 필요한 라이브러리 루틴으로 생성하였다.

3.3.2 NMS 인터프리터

네트워크 관리 스크립트 인터프리터 모듈은 스크립트 언어의 구문을 검사하는 기능을 제공하는 루틴

으로서, 입력된 스크립트의 문법적인 오류가 있을 경우 이를 스크립트 관리 서비스 사용자에게 오류 상황을 통지해 주며, 오류가 없을 경우 스크립트의 수행을 위한 스크립트 수행기(script executor) 모듈로 관리 스크립트를 넘겨주는 역할을 담당한다. NMS 인터프리터는 먼저 사용자 인터페이스를 통하여 받아들인 정보를 이용하여 관리 스크립트를 구성하고, 그 다음 단계로 작성된 관리 스크립트의 문법적 오류를 검사한다. 오류 검사를 위하여 어휘 분석기를 구성하기 위한 lex와 LALR(1) 파서 생성기인 yacc를 이용하여 NMS 인터프리터 모듈을 작성한다. 오류가 없는 관리 스크립트는 이후의 관리나 재사용을 위하여 스크립트 테이블에 저장하게 되는데, 스크립트 테이블의 검색 및 등록 등을 수행하는 스크립트 테이블 관리 루틴(STH routine)을 호출한다. 한편 컴파일러 자동화 도구를 이용하여 NMS 인터프리터를 작성함으로써 스크립트 문법을 확장하거나 변경하고자 할 때 문법 규칙과 각 규칙에 대한 수행 코드만을 변경하면 되므로 확장이 용이하고, 각 규칙이 reduce될 때 수행하는 코드는 CMIP machine에서 제공하는 기능들을 직접 호출하는 형태로 작성하였다. 이 때 시간에 관한 정보는 그대로 스크립트 수행기 모듈로 전송되게 된다.

3.3.3 STH 모듈과 관리 정보 베이스

스크립트 테이블 관리 루틴에서는 스크립트에 관한 정보를 검색하고, 등록하는 일을 수행하므로써 스크립트의 재사용 및 갱신, 위임 사항 등에 관한 모든 정보를 보관, 관리하는 역할을 담당한다. 스크립트 테이블의 관리는 해쉬 심볼 테이블을 이용하여 관리하며, 관리 객체 클래스별로 관리가 가능하도록 객체 클래스에 관한 정보를 표현하고, 스크립트명을 심볼로 사용하였으며, 위임 정보 등을 기록하게 된다.

스크립트 테이블에서 스크립트명은 스크립트 작성 시 부여하는 스크립트의 이름이며, 스크립트가 작성된 시기와 작성자가 표에 기록되고, 관리 대상이 되는 객체에 대한 정보와 위임에 관한 정보가 기록된다. 관리 객체에 대한 정보는 OSI에서 권고한 의미와 동일하고, 위임에 관한 정보는 먼저 위임 상태는 정수로 표시되는데, 0은 위임이 되지않고 정상적으로 관리자가 관리 스크립트를 수행하고 있음을 나타내며, 1은 위임된 상태, 2는 위임된 스크립트의 수행 중지(suspend) 상태를 표시하고, 3은 위임된 스크립트의 내용 수정을 하고 있는 상태를 표시한다. 이 중 상

태 2는 긴급한 상황의 발생으로 스크립트 수행을 중지한 상황을 표현하고, 필요에 따라 수행을 재개할 수 있으며, 수행이 재개되면 상태는 위임을 나타내는 1로 바뀌게 된다. 스크립트 포인터는 작성된 스크립트가 저장되어 있는 기억 장소를 포인터한다.

스크립트 테이블 관리 루틴은 NMS 인터프리터의 요청을 받아 테이블에 하나의 엔트리를 생성한 후 콘트롤을 NMS 인터프리터에게 되돌려 주고, 관리 스크립트의 수행시 스크립트 수행기의 요청에 따라서 테이블 내의 위임 상태를 갱신하게 된다. MIB를 구축하기 위해서 관계 데이터 베이스 형태의 실험적인 데이터를 사용하였으며, ISODE와 관련된 isobjects, isoentities, isoservices 부분에 정보와 스크립트 관리를 위한 화일을 첨가하였다.

3.3.4 스크립트 수행기와 타이머

스크립트 수행기(script executor) 모듈은 NMS 인터프리터가 스크립트의 오류 검사 후 생성한 CMIS 수행 루틴들과 시간 정보를 받아 네트워크 관리를 위한 스크립트를 수행하는 루틴으로서, 스크립트 관리 서비스 사용자로부터 위임에 관한 정보를 입력받아 입력된 정보에 따라 스크립트를 수행하게 된다. 첫째 위임이 없을 경우에는 관리자 측에서 스크립트를 수행하게 되며, 먼저 관리자측의 local MIB를 찾고, local MIB에 존재하지 않을 경우 피관리자와 연결을 설정한 후 원격 동작을 이용하여, 피관리자 관리 정보 베이스로부터 필요한 정보를 얻거나 수정할 수 있다. 둘째 위임에 관련된 상황을 처리할 경우에는 위임의 종류에 따라 스크립트 테이블을 갱신하고, 위임일 경우 해당 스크립트를 피관리자 측으로 전송한다. 전송된 스크립트의 수행은 피관리자의 책임하에서 수행되며, 관리자는 단지 스크립트 테이블을 참조하여 위임된 스크립트의 수행 중지나 재개, 갱신을 요청할 수 있다. 또 스크립트의 수행을 위해서는 반드시 시간 정보에 따라야 하며, 제공된 시간 정보에 따라 스크립트를 수행하게 되는데, 수행시간이 되지 않은 스크립트는 큐에 보관하게 되고, 수행 시간이 지나버린 스크립트는 이를 스크립트 관리 서비스 사용자에게 오류 메시지와 함께 통보한다.

타이머(timer) 모듈은 스크립트의 시간에 관한 정보를 처리하기 위한 루틴으로서 스크립트의 시간 정보를 해석하여, 자원 감시 및 스크립트 수행을 통제하는 역할을 담당한다. 이는 스크립트 수행 프로세스의 시작과 종료, 중지나 재개 등을 통제하게 된다. 타

이머는 델타 리스트(delta list)로 구현하였으며, 타이머 프로세스는 스크립트 수행기 프로세스와 IPC (Inter Process Communication)를 통해 통신하게 된다. UNIX에서 지원하는 다양한 IPC메커니즘 중 사용이 간단하고 확장성과 이식성이 뛰어난 메시지 큐(message queue) IPC를 선택하였으며, 타이머 프로세서는 IPC를 초기화하고, IPC를 통해 들어오는 사건을 받아 델타 리스트에 노드를 생성한 후, 시간으로 표현되는 key 필드 값을 줄여 나간다. 타이머 노드의 key필드 값이 '0'이 되면 델타 리스트로부터 노드를 삭제하고 이를 스크립트 수행기에게 통지함으로써 스크립트의 주기적인 수행을 가능하게 할 수 있다.

IV. 결과 및 고찰

본 장에서는 스크립트 기능을 이용한 네트워크 관리 예를 보인다. 관리자가 피관리자 시스템의 관리 객체인 프린터를 관리하는 경우를 고려하면 피관리자 시스템의 프린터에 오류가 발생했을 경우 관리자는 사용자에게 계속적인 서비스를 제공하기 위하여 프린터의 상태를 감시한다. 감시 중 오류 발생이 감지되면 관리 정보 베이스의 상태값이 준비로 되어 있는 다른 프린터를 찾아 그 이름으로 변경하여 준다. 따라서 사용자는 프린터의 오류 상태와 관계없이 계속해서 서비스를 받을 수 있다. 이를 관리 스크립트 언어로 기술하면 그림 5와 같다.

```

/* this is a simple script for printer management */
script print_control :
begin
/* examine printer's status */
on status = "fail" :
/* find another available printer */
test(class,status);
if status = "ready" then
begin
/* recover from failure */
recover set(class,instance,attribute) ;
/* notify to manager with message */
notify(message) ;
end
end
/* check status at ten hours AM during one hour every 30 seconds */
at 16:10:00:00 for 16:11:00:00 every 00:00:30
    
```

그림 5. 프린터 관리 스크립트 예

Fig. 5. An example for printer management script

그림 5의 스크립트는 관심있는 관리 객체인 프린터의 상태를 매 30초 마다 16일 오전 10시 부터 11시 까

지 한시간 동안 검사하여 오류 상태가 발견되면 다른 사용 가능한 프린터로 출력할 수 있도록 처리하고 이를 관리자에게 통보하는 스크립트이다. 일단 작성된 스크립트는 NMS 인터프리터가 구문 오류를 검사 후 스크립트 테이블에 필요한 정보를 수록하고 이를 CMIP 프리미티브로 변환하여 준다. 그림 5의 관리 스크립트는 그림 6과 같은 형태로 변환된다.

```
#include <stdio.h>
#include "CMIP_types.h"
#include "CMIP_error.h"

main()
{
  /*(status == "fail") */
  M-GET(m_get);
  if (m_get->mgntInfoList->MgntInfo->element_CMIP_7->param == 1)
  {
    M-GET(m_get);
    if (m_get->mgntInfoList->MgntInfo->element_CMIP_7->param == 2)
    /*(status == "ready") */
    {
      M-SET(m_set);
      M-EVENT-REPORT(type_CMIP_EventReportArgument);
    }
  }
}
/* at 16:10:00:00 for 16:11:00:00 every 00:00:00:30 */
```

그림 6. NMS 인터프리터의 수행 결과
Fig. 6. result of interpretation

그림 6은 NMS 인터프리터가 그림 5의 관리 스크립트를 읽어 처리한 결과이다. 주석문의 처리 및 해당 CMIP 프리미티브를 생성하며, 오류가 없을 경우 결과를 스크립트 수행기로 전달한다. 이 때 관리자는 정보 전송량을 고려하여 위임 여부를 결정하게 된다. 위임에 관련된 상황을 처리할 경우에는 서비스의 종류에 따라 스크립트 테이블을 갱신하고, 각 서비스에 따른 처리를 관리 알고리즘에서 기술한 바에 따라 수행한다.

한편 스크립트 수행기는 스크립트의 수행을 위해서 시간 정보를 분석하는데 at절에서 기술된 시간이 이미 지나버렸을 경우는 관리자에게 오류를 통보하

고, 현재 시간과 일치할 경우 스크립트 수행을 개시한다. 이 때 타이머 프로세스를 생성하고 주기적인 수행을 위해 델타 리스트에 노드를 첨가한다. 위에서 타이머 노드 구조는 그림 7과 같다.

그림 7에서 스크립트 수행 프로세스의 PID가 100이라 하면 매 30초 마다 IPC를 통하여 스크립트 수행 프로세스에 메시지를 전송한다. 스크립트 수행기는 이 메시지를 읽고 프로세스를 통제하게 된다.

앞에서 살펴본 바와 같이 스크립트를 이용한 네트워크 관리를 수행함으로써 예측되는 장애 상황의 탐지나, 오류 복구, 통지 등을 자동적으로 수행할 수 있는 장점을 가진다. 한편 이러한 관리 스크립트는 작성 후의 재사용이나 위임 등을 결정할 수 있으므로 관리자 시스템의 부담을 줄일 수 있고 신뢰도도 높게 된다. 네트워크 운영자나 관리 책임자 등이 네트워크를 관리하고자 할 경우 이에 필요한 명령어는 command나 shell 등을 통하여 네트워크 관리 시스템에게 요청한 후, 시스템으로 부터의 응답을 기다리는 형태로 표현되고 있다. 그러나 이러한 방법은 수많은 네트워크 정보 중 관리자에게 필요한 정보만을 선택하여야 함은 물론 이를 분석한 후 적절한 관리 행위를 결정하여 관리 시스템에 다시 명령해야 하는 등 관리자의 부담을 증가시키고, 필요한 회복 조치의 지연 등으로 네트워크에 중대한 영향을 미칠 수도 있다. 하지만 네트워크 내에서 발생하는 대부분의 문제점은 예측 가능한 것이고, 예측 가능한 문제에 대한 해결책은 사전에 제시할 수 있으므로 이러한 예측성을 이용한 네트워크 관리가 가능하게 된다. 특히 관리 스크립트 언어 설계시 시간 개념을 첨가하여 스크립트 작성시에 관심있는 관리 대상을 설정하여 이를 정해진 시간 동안 주기적으로 감시하는 기능을 부가함으로써 관리 대상의 특성에 따라 정해진 시간에 관리 동작을 유발시키거나, 관리 동작을 일정시간 지연시키거나, 일정한 시간 간격을 두고 관리 동작을 반복하는 등의 시간 개념을 기술할 수 있도록 하였다. 따

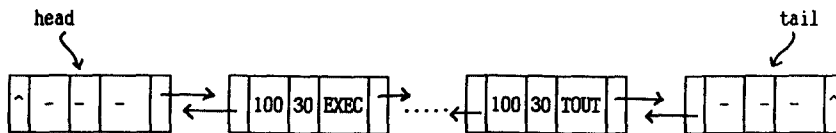


그림 7. 타이머 노드 구조
Fig. 7. delta list representation for timer

라서 관리 대상의 특성 및 긴급도 등을 반영할 수 있는 특징을 지니게 된다. 스크립트 관리 기능이 추가됨으로 인하여 네트워크 운영자나 오퍼레이터 등의 부담이 감소됨은 물론 네트워크 장애의 조속한 발견과 장애로 부터의 신속한 복구 등이 이루어지는 장점을 지니게 된다. 더불어 이기종 네트워크 시스템 관리시에 발생하는 문제점인 다양한 관리 정보 교환 프로토콜에 대해 별도의 응용 프로그램을 작성하지 않고, 관리 스크립트 인터프리터 만을 수정하면 가능함으로 보다 높은 적응성을 지니게 된다.

V. 결 론

위임 모델은 관리자 와 피관리자 간에 관리 기능을 필요에 따라 분배하는 OSI 네트워크 관리 모델의 확장된 개념으로 볼 수 있다. 이 모델에서는 위임을 통하여 관리자 와 피관리자 간에 관리 기능을 적절하게 분배함으로써 교환되는 관리 정보량을 최소화할 수 있도록 설계하였고, 피관리자 측면이 아닌 관리자 측면에서 관리 스크립트를 표현할 수 있도록 하였다. 본 논문에서는 간단한 관리 스크립트 언어를 설계하고, 이를 위한 인터프리터를 구현하였다. 또 효율적인 네트워크 관리를 위해 스크립트 관리 기능과 서비스를 정의하였으며, 관리자 시스템과 피관리자 시스템 모델을 설계하였다. 그리고 이를 지원하기 위한 관리자 알고리즘과 피관리자 알고리즘을 제시하며 구현을 통하여 그 기능을 확인하였다. 앞으로 스크립트 관리 프로토콜의 파라미터들에 대한 분석과 선택에 관한 계속적인 연구가 뒤따라야 할 것이며, 각 서비스에 대해서도 개선하여야 할 많은 부분을 포함하고 있다. 더불어 규칙 기반 개념을 관리 스크립트 언어에 도입하는 문제와 이를 위한 인터프리터에 대한 연구가 계속되어야 할 것이고, ISO 표준안을 완전하게 따를 수 있는 관리 정보 베이스의 구축이 이루어져야 할 것이다.

참 고 문 헌

1. Y.Lamdan, B.Iyer, G.Barzilai, A.Cahana, "Dialog Manager for Network Management," Proceedings of the IFIP TC6/WG6.6 International Symposium on Integrated Network Management, 1989, pp.269-278.
2. ISO/IEC DIS10040, Systems Management Overview, 1990.
3. OSI/Network Management forum Architecture, Forum 004, issue1, January 1990.
4. Yechiam Yemini, et al., "Network management by delegation," Proceedings of the IFIP TC6/WG6.6 Second International Symposium on Integrated Network Management, 1991, pp. 95-107.
5. Yuka Kamizuru, et al., "A Proposal for a Network Management Model on the Widely Integrated Distributed Environment," Proceedings of 6th International Joint Workshop on Computer Communications, 1991, pp.319-324.
6. ISO/IEC DIS 10165-1, Structure of Management Information-Part 1: Management Information Model.
7. ISO/IEC IS 9596-2, Management Information protocol specification-Part 2: Common Management Information Protocol, 1989.
8. David Holden, "Predictive languages for management," Proceedings of the IFIP TC6/WG6.6 International Symposium on Integrated Network Management, 1989, pp.585-596.



韓 順 姬(Soon Hee Han) 正會員
 1960年 8月 9日生
 1983年 2月:慶北大學校 電子工學
 科學士
 1985年 2月:光云大學校 大學院 電
 子計算學科 碩士
 現在:光云大學校 大學院 電子計算
 學科 博士課程 修了

1989年~現在:國立麗水水產大學 컴퓨터工學科 助教授

조 국 현(Kuk Hyun Cho) 正회원
 1977년:한양대학교 공과대학 전자공학과 졸업(학사)
 1981년:일본 동북대학교 대학원 전자통신공학과 졸업(석사)
 1984년:일본 동북대학교 대학원 전자통신공학과 졸업(박사)
 1984년~현재:광운대학교 교수
 1987년~1991년:한국정보과학회 정보통신연구회 총무, 운영위원 역임
 1991년~현재:한국정보과학회 정보통신연구회 분과 위원장
 ※관심분야:네트워크 관리, 프로토콜 공학, 지능 망, 실시간 처리 성능 평가, 관리 언어



이 재 오(Jae Oh Lee) 正회원
 1987년:광운대학교 이과대학 전자계산학과 졸업(학사)
 1989년:광운대학교 대학원 전자계산학과 졸업(석사)
 1989년~현재:광운대학교 대학원 전자계산학과 박사과정
 ※관심분야:네트워크 관리, 프로토콜 공학, 지능 망, 실시간 처리