

## 論 文

# Ada언어를 이용한 통신 프로토콜의 명세화에 관한 연구

正會員 金 吉 俊\*

## A Study on the Specification of Communication Protocols Using Ada

Gil Choon KIM\* Regular Member

**要 約** 본 논문은 Ada의 주요개념과 OSI 전송을 위한 전송모델의 구조 및 가능을 알아본 후 전송 인터페이스를 Ada 언어로 기술 가능함을 제시하였다. 이는 Ada언어를 사용하여 통신 프로토콜을 명세화하고자서 통신 프로토콜처리용의 프로그램 생성이나 프로토콜 세부의 작성을 Ada언어로 수행될 수 있도록 함으로서 소프트웨어 검증등에 좋은 효과를 제공하기 위함이다.

**ABSTRACT** In this paper, in first we show a main concept of Ada language and the architectures and functions of transport model for OSI, and than represent the possibility of describing the transport interface using Ada language. It is given a following result. As we specify communication protocol using the Ada, we can to create of program for processing the communication protocol or for processing protocol itself. It is known to us to represent the effect debugging of software.

## I. 서 론

Ada가 통신 프로토콜 분야에서 중요한 이유로서는 병렬현상과 응용문제에 대한 모듈화가 가능하다는 점이다. 모듈성은 task 단위와 패케이지 단위로 표현되는데 이 패케이지는 objects, operators, functions, procedures로 구성되므로 이를 모듈에 참가시키므로서 abstract data types를 기술할 수 있다. 이러한 기능과 구조(그림 1)를 패케이지 및 procedure의 표시와 추상자료의 기술을 가능케하는 Ada로 쉽게 표현할 수 있다.

프로토콜 layer에서 필요한 data type을 다루기 위해서는 data 기술과 여러 function을 data type상에서 분할과 조화를 해야하고, sdu(service

data unit), pdu(protocol data unit), sdu(service data unit)를 위한 data 정의도 필요하다. 만약 이때 모든 상태에 영향을 미치는 특별한 함수인 경우는 별도로 고려해야만 한다.

concurrent 문제는 task와 Rendez Vous(RV)과의 개념으로서, task 기술은 task 상에서 task type과 access값의 기술로 쉽게 다루어지며, task 사이에 메세지를 전달하는 통신기기인 RV는 프로토콜(protocol machine)과 사상기계(mapping machine)사이에 pdu를 전달하면서 프로그램에서 하나의 Procedure call처럼, 호출 task에 값을 Return한다.

Protocol machine은 data 전송시 도달여부를 점검하고, 전송 error시 재전송을 위한 정보를 각 층의 entity(통신이 가능하도록 하는 능동적 요소)간에 주고 받기도 한다. 또한 이를 위한 정보의 내용과 형식은 각 층마다 정해져 있다. mapping machine은 (network) address상에서

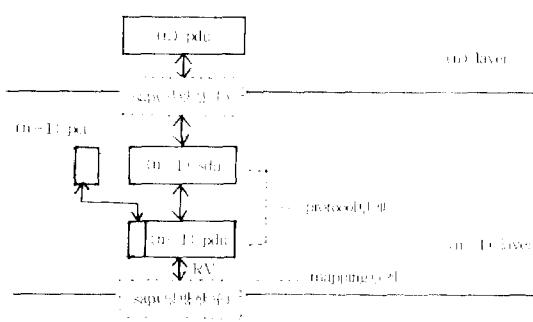
\* 光州大學校 電子計算學科

Computer Science Department of Kwang Ju University  
論文番號 : 91-64 (接受1990. 9. 20)

전송등 address를 mapping하고, multiplexing과 그 실행을 결정하며, (network) service의 관리등을 하는 기능을 가지고서 모든 protocol network을 단절하고 그 관리자의 다른 network 등을 일치시킨다. 또한 protocol과 task는 mapping하기고 ressource controller로서 task mapping을 관리하거나 삽입을 한다.

protocol과 mapping machine은 RV를 통하여 상호처리를 도우며 mapping machine의 waiting queue는 두 machine 사이의 미동기를 가능케 한다.

그리고 data transfer의 효율을 증가시키기 위해 몇 가지 통신 규칙을 설정하고 있으며, protocol machine은 calling task로서 3개(pdu 전송, pdu 수신, 재설정용 제어 명령)의 RV로서 data는 access 한다.



\* 하위망상은 데이터 전송죽을, 상위망상은 데이터 전송수단을  
보석  
\* [] : 전송데이터 단위, → : 데이터 흐름, RV : Rendez Vous

그림 1. 프로토콜 인터페이스의 scheme  
Fig. 1. Scheme of protocol interface

여기서 (n) layer의 전송 data의 단위인 (n)-pdu 을 (n-1) 계층에 전송의뢰하면 (n-1) layer의 (n-1)-sdu로서 인식한다. 또한 (n-1) layer에서는 (n-1) 프로토콜로 필요한 사이 정보인 (n-1) pci(protocol control information) 를 첨가한 (n-1)-pdu를 다시 하위 계층으로 전송을 한다.

## II. OSI모델 표현과 Ada

OSI 전송모델을 전송국 프로그램으로 작성하기 위해 사용자들은 자체의 복잡성을 줄이기 위한 가능(load, link)을 group으로 분리한 후 layer의 모듈을 정의한다.

이 시스템 전송국은 프로세스들의 집합으로서 전송서비스를 제공한다. 전송인터페이스는 sap(상호작용 인터페이스)와 서비스를 실현시키고 이를 sap type과 그 procedure 정의를 Ada로 표현하여 가능하며 그 인터페이스 procedure는 3부분(예, transport service data unit인 경우)으로 나뉘어 기술될 수 있으며 여기서 각 sap은 영향 대역이나 송수신을 효율적으로 수행되도록 한다.

- 인터페이스를 위한 procedures 기능요약  
- sap 관리 절차 : 이를 위한 사용자에게 sap를 참조 또는 관리한다.

- 통신 절차 : 이를 위한 사용자에게 전송프로토콜을 사용하도록 한다.

- 들어오는 전송연결을 사용자에게 통보하는 절차 :

예를들면, 사용자가 전송연결을 다시 open하고 사용한다면 다음과 같이 기술해야 한다.

관리 절차(c) :

```
if c<0      return code로서 sap id의 값 0을
    사용
```

than 통신 절차(c,tconreq) :

```
else 다른 내용 처리      더 이상 sap 유호하지
    않을지
```

end if :

session이나 sap을 초기화한 방법으로는 자원관리자를 이용하여 sap관리 절차를 수행시킨다. 그래서 하나의 critical session처럼 sap을 다루어서 전송인터페이스에 requests의 충돌을 해소된다. 이러한 충돌은 연결이 단절된 동안에 이드의 sap에서 일어나는 것으로서 T-DISC-REQ와 T-DISC-IND가 동시에 sap에서 관리하는 경우이다. 환경 세이브를 각 sap에 의해 제공되며

입력되는 sdu은 사용자가 통신절차로 읽을 때까지 sap내에 저장시킨다. 그리고 흐름제어는 전송과 session 절차 사이에서 자동적으로 수행되도록 한다. 이는 통신절차가 동기식이나 비동기식에서 사용되어질 수 있다는 것을 알 수 있다.

즉, 사용자는 자기가 sap내에 어떤 것이 있을 때 까지 기다리거나 자기 자신의 코드를 검색하여 그 자신의 코드로 돌아가든지의 여부를 선택할 수 있다. 이 선택은 Boolean 변수로서 해결될 수 있으며 보다 좋은 해는 대기시간을 나타내는 파라메터를 사용해야 한다.

다음은 Ada개념이 이러한 형태로 사용된 RV이다.

```
select
  accept readsap(sdu) ;
  or
  delay x ;
end select ;
```

프로토콜기체는 전송프로토콜을 수행하는 기계로서 sap을 통하여서는 session layer와 메세지를 교환하고, RV에 의해서는 사상기체와 메세지를 교환한다.

하나의 모니터는 상호배제시에 할당과 비할당 기능들을 제공하기 위해 모든 task를 관리한다. task의 동적 할당으로 active 프로토콜 task의 번호와 active connection의 번호는 똑같게 되며, 다음과 같은 task의 동적 할당으로 이루어지고 전송연결은 concurrent하게 진행한다.

#### 관리절차(sap)

dynamic allocation of a sap

dynamic allocation of } done by with a monitor protocol

initialization of protocol

end 관리절차 ;

프로토콜 task의 내부구조는 FSM(Finite state machine)의 프로그램이며 통신의 내용은 지역(Local) 부분에서 선언된 변수들이다. 또한 그 predicate은 이 내용을 사용하는 Boolean 함수들로서 그 전이는 case문으로 표현된다.

```
통신절차(sap, ph, sdu, sync) : -- read a tsdu in the
                                associate sap
code=decode(sdu) :
  case code is
    when T-CON-req => case currentstate is
      when CLOSED => if p1 then ...
                    alloc-path(p,r,par)
      :
      :
      elseif ... end case ;
    end case ;
```

task은 각각 데이터의 분할과 조합을 수행하는 두부분으로 나누어지며 분할부분은 sap을 읽고, 조합부분은 입력되는 데이터를 위한 path를 읽는다. 동적 할당의 task 모듈이 갖는 하나의 문제점을 살충단계시 프로토콜 class의 switch를 허락하는 것으로서 간단한 해는 그 프로토콜 task내에 모든 class들을 놓아 그들을 predicate처럼 내부 switch와 분리하는 것이다. class 1,3,4에서 time 제약들을 나타내기 위해 시는, timer을 transition에 조합시켜 프로토콜 task에 링크시킨다.

```
task timer is
  entry arm(x : duration) :
    entry stop ;
  end timer ;
  task body timer is
    y : duration
  begin
    loop
      accept arm(x : duration) do -- arm the timer
        y := x ;
      end arm ;
      select
        accept stop : -- waiting the stop
        or
        delay y : dest.expir ; -- signal
        end select ;
      end loop ;
    end timer ;
```

다음은 Ada구조와 OSI의 구조의 관계를 나타낸다.

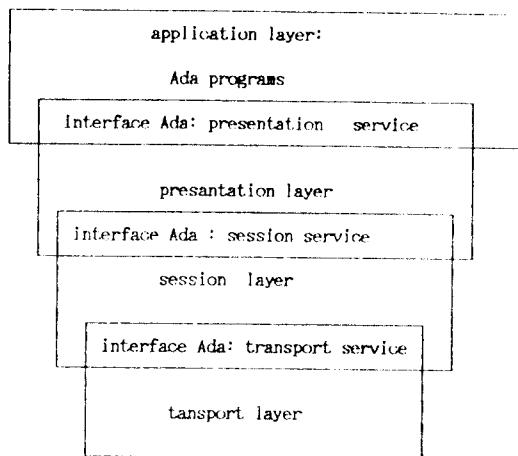


그림 2. Ada와 OSI 구조  
Fig. 2. architecture of Ada and OSI

Ada 인터페이스는 그것을 다음은 communication들과 function들의 명세화로 사용하게 하는 Ada 메세지가 될수 있으나 또한 데이터 캡슐화도 가능하다.

### III. 결 론

이 논문은 통신 프로토콜분야에서 명세화언어로서 Ada의 적합성을 알아보았다. 여기서 명령과 실시간이라고 하는 점에서 프로토콜의 명세화와 Ada 언어의 동질성은 다른 언어에 비해 중요한 장점이다. 그러나 구현하는 면에서 볼때 performance가 조금 뒤떨어진다(ex, C 언어). 결론적으로 통신프로토콜의 명세화는 소프트웨어화의 데이터 캡슐화로서 소프트웨어 검증에도 좋은 효과를 제공한다.

그래서 최근에는 통신프로토콜처리용 프로그램이나 프로토콜 제품의 작성을 위해 프로그램언어

와 친화성이 높은 프로토콜의 형식기술언어(LOTOS, SDL등)에 대한 연구나 표준화가 활발히 진행되고 있다. 그러나 형식기술언어 이외의 다른 언어인 Ada로도 효율적으로 수행될 수 있음을 프로토콜 표준화에 많은 도움이 되리라 생각한다.

### 참 고 문 헌

1. D.Dykeman and W.BUX, "Analysis and tuning of the FDDI media access control protocol", IEEE JSAC, Vol.SAC-6, no.6, 7 1988
2. F.Halsall, "Data communications, computer networks and OSI", Addison wesley, New York, 1988
3. G.Andreoni, "Fortran interface to X.25 and transport service computer networks", 8 1984
4. R.J.A. Buhr, D.A. Mac Kinnon, "The transport layer in OSI Research report DOC CR CS -1980 -008 prepared for the department of communication of Canada", 1981
5. Stallings, William, "Data and Computer Communications", Macmillian Publishing Company, 1985



金 吉 俊(Gil Choon KIM) 正會員  
1949년 4월 25일 生  
1975. 1 : 충남대학교 공과대학 전자계산  
학과 졸업(공학사)  
1981. 8 : 충남대학교 대학원 전자계산학  
과 졸업(공학석사)  
1990. 2 : 충남대학교 대학원 박사과정  
전자계산학과 주교(시스템  
소프트웨어전공)  
1985. 5 ~ 현재 : 광주대학교 공과대학  
전자계산학과 부교수