

UCA(Utility Communication Architecture)와 필드버스를 이용한 전력설비 자동화통신망 구축 방안 연구

황인휘* 최인호* 홍승호*
† 한양대학교 제어계측공학과 ‡ 한양대학교 전자컴퓨터공학부

Study on the implementation of power utility network using UCA and Fieldbus

In Hui Hwang* In Ho Choi* Seung Ho Hong‡

† Graduate School of CIE, Hanyang University ‡ School of EECS, Hanyang University

Abstract - 본 연구에서는 UCA와 필드버스를 이용한 전력 설비 통합 자동화통신망 구축 방안을 제시한다. 이를 위하여 (i) UCA와 필드버스의 프로토콜간 인터페이스 작업 최소화화, (ii) 게이트웨이 내에서의 연산작업 최소화의 요구사항을 만족하는 UCA-필드버스 게이트웨이 구성 방안을 제시하고, 게이트웨이 내에서 UCA와 필드버스의 프로토콜 매핑 방법을 기술한다.

1. 서 론

지난 수년간 컴퓨터와 통신 기술이 급격히 발전함에 따라 전력 설비에 정보화 기술을 기반으로 하는 자동화 시스템의 도입이 매우 활발히 추진되고 있다. 미국의 EPRI에서는 1988년부터 CASM[3], GOMFSE[4] 및 TASE.2[5]로 구성된 전력 설비용 개방형 통신 프로토콜인 UCA[1-7]의 개발에 착수하였다. 이중 CASM은 전력 설비의 필드 장비 객체 모델에 공통으로 사용되는 통신 서비스를 정의하고 있으며, 따라서 UCA와 필드버스의 결합은 CASM 프로토콜을 필드버스와 매핑시킴으로써 이루어질 수 있다.

UCA의 CASM 프로토콜은 필드버스의 응용계층 및 사용자계층 프로토콜과 매핑되어야 한다. 본 연구에서는 다음의 두 가지 요구사항이 만족될 수 있는 UCA-필드버스 게이트웨이 구성 방안을 제시한다.

- UCA와 필드버스의 프로토콜간 인터페이스 작업 최소화
- 게이트웨이 내에서의 연산작업 최소화

본 연구에서는 현존하는 여러 종류의 필드버스들 가운데 유럽지역에서 개발되어 현재 전세계적으로 가장 높은 시장 점유율을 보이고 있는 PROFIBUS[8-9]와 최근에 미주지역에서 개발이 완료되어 앞으로 널리 사용될 것으로 예상되는 FOUNDATION Fieldbus[10-11]를 대상으로 UCA와 필드버스의 인터페이스 방안을 제시한다.

2. 본 론

발전소와같은 전력설비에는 설비내 기간 통신망으로 LAN이 설치된다. 이러한 경우 필드 디바이스들간의 통신을 담당하는 필드버스는 전력설비 LAN과 접속되어야 하며, 원격지의 제어센터와 전력설비간의 통신은 전력설비 LAN과 제어센터 LAN을 연결하는 WAN을 통하여 수행되어야 한다. 그림 1에는 이러한 전력 설비 통신망의 물리적 구성이 나타나 있다. 그림에서 보는 바와 같이 전력설비의 LAN과 필드버스를 접속시키기위해

LAN-필드버스 게이트웨이를 필요로 한다.

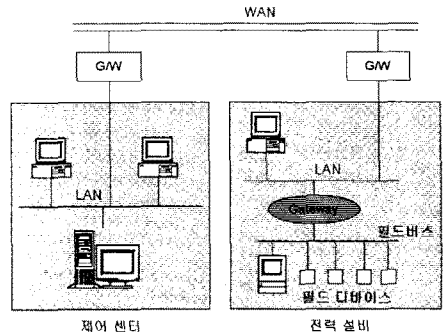


그림 1 전력설비 통신망의 물리적 구성

이렇게 구성된 시스템에서 발전소의 전력설비와 제어 센터간의 실시간 데이터 교환을 위하여 UCA의 TASE.2 프로토콜을 필요로한다. 또한 원격지의 제어센터에서 전력설비 디바이스들을 실시간으로 제어하기 위해 제어센터의 LAN과 전력설비의 LAN에는 UCA의 CASM을 사용한다. 따라서 그림 1에서 제시한 전력설비 디바이스와 제어센터간의 통신은 그림2에 나타난 것과같은 UCA CASM과 필드버스를 인터페이스시킴으로써 이루어진다.

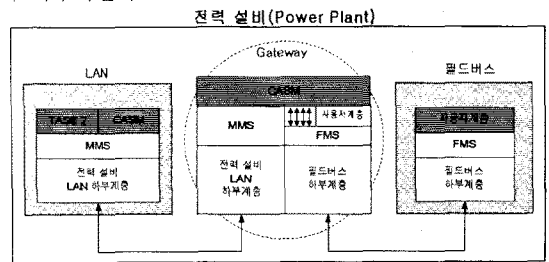


그림 2 전력설비 통신망에서 프로토콜간의 인터페이스

그림 2의 구조에서 제어센터와 전력설비들 간의 통신을 담당하게 될 CASM이 제공하는 통신 서비스는 필드 디바이스와의 통신을 위해 기존의 MMS대신 필드버스의 응용계층인 FMS 서비스 및 필드버스 사용자 계층 서비스를 이용하여 수행되어야 한다. 따라서 UCA와 필드버스의 인터페이스 구현에 있어 CASM과 필드버스 응용계층 또는 CASM과 필드버스 사용자 계층간의 매핑작업이 요구된다. 이로써 UCA와 필드버스간의 인터페이스에 요구되는 작업을 최소화하는 게이트웨이를 구현할 수 있다.

제어센터의 사용자계층에는 전력설비와의 실시간 데이터 교환을 담당하는 TASE.2가 탑재되며, 제어센터와

본 연구는 1998년 한국전력공사 지원에 의하여 기초전력공학공동연구소 주관으로 수행되었음(과제번호 98-080)

전력설비의 인터페이스를 위하여 TASE.2와 CASM의 인터페이스 작업이 요구된다. 이러한 TASE.2와 CASM의 인터페이스는 UCA-필드버스 게이트웨이 내에서도 수행될 수 있으나, 이 경우 게이트웨이에서 과도한 연산작업을 초래하게 되므로 그림 2에서와 같이 TASE.2와 CASM의 인터페이스가 제어센터의 컴퓨터 내에서 처리되도록 하여 게이트웨이 내에서의 연산작업을 최소화하는 것이 바람직하다.

필드버스는 응용계층 외에 별도의 사용자계층을 정의하고 있으나 UCA에서는 응용계층과 사용자계층이 명확히 구분되어있지 않기 때문에 UCA의 CASM을 필드버스의 응용계층 또는 사용자계층의 어느 한쪽으로부터 매핑시키는 것은 타당하지 않다. 따라서 UCA와 필드버스의 인터페이스는 (i) 필드버스의 응용계층으로 직접 매핑이 가능한 부분, (ii) 응용계층에는 직접 매핑되지 않으나 사용자계층에 매핑 가능한 부분, (iii) 응용계층과 사용자계층 양쪽 어느 계층으로도 매핑되지 않아 필드버스 측에서 추가적으로 서비스를 구현하여야 하는 부분으로 구분하여 수행되어야 한다. 본 논문에서 대상으로 하는 FOUNDATION Fieldbus와 PROFIBUS-PA는 모두 응용계층 프로토콜로 FMS(Fieldbus Message Specification)를 사용한다.

UCA의 CASM에는 LogicalDevice, DataObject, DataSets, FunctionalComponent등의 디바이스 모델과 Server, Association, Report, Device Control, Multicast, Time등의 서버 모델이 있으며, 이러한 UCA와 필드버스 간의 매핑은 각각의 모델들이 제공하는 다양한 오브젝트와 서비스에 대해 모두 수행되어야 한다. 본 연구에서는 이러한 모든 오브젝트와 서비스에 대한 매핑방법을 제시하였으며 본 논문에서는 수행된 UCA와 필드버스의 매핑에 대해 대표적 예만을 기술하기로 한다. 우선 필드버스의 응용계층 직접 매핑의 대표적인 예로 CASM의 DataObject 디바이스 모델의 매핑 테이블을 통해 CASM-FMS 매핑 방법을 기술한다. DataObject 모델은 필드 장비 데이터와 모델에서 특별히 요구되는 데이터들에 대한 구성과 이름을 표현하는 요소이다. 다음의 표 1에는 CASM의 DataObject 모델과 FMS의 Named Variable간에 객체 속성(Attribute) 및 통신 서비스(Service)의 매핑 관계가 나타나 있다.

표 1 CASM-FMS의 DataObject 모델 매핑

Object Class	FMS Class	Server	Client
DataObject	Named Variable	m	m
Attribute	FMS Object/Attribute		
DOReference	Simple Variable Name	m	m
Ancestry	visible-string	m	o
Scope	DOMAIN, VMD, CM		
DataType	Type Specification	m	m
Value	Data	m	m
AccessControl	Access Method		
Deletable	Deletable		
LocalResource	Address		
Services	FMS Services		
GetDataObjectValues	Read	m	m
SetDataObjectValues	Write	o	o
SetDataObjectValues (Unconfirmed)	InformationReport	o	o
GetDataObjectAttributes	ReadWithType	m	m
CreateDataObject	DefineNamedVariable	o	o
DeleteDataObject	DeleteVariableAccess	o	o

(m: mandatory, o: optional)

표 1에 나타난 FMS의 서비스 가운데 이텔릭체로 표시된 부분은 CASM이 요구하는 통신 서비스를 FMS에서 제공하지 못하는 경우로 이러한 서비스들은 추가로 구현되어야 한다. 본 연구에서는 CASM과 MMS간에 매핑 관계(3)를 참고로 하여 CASM에서 제공하는 서비스를 FMS에 추가시키는 방법을 제시한다.

다음에는 FMS에서 추가로 구현되어야 하는 서비스의 예로 CASM DataObject 모델의 CreateDataObject 서비스와 매핑하기 위한 추가적인 서비스의 구현 방법을 기술한다. 다음의 표 2와 표 3에는 각각 CreateDataObject 서비스와 매핑시키기 위해 정의한 DefineNamedVariable 서비스의 ASN.1 정의와 프리미티브를 나타내었다.

표 2 DefineNamedVariable의 ASN.1 정의

ASN.1 정의	
DefineNamedVariable-Request ::= SEQUENCE {	
VariableName	(0) ObjectName
address	(1) Address
typeSpecification	(2) TypeSpecification OPTIONAL
}	
DefineNamedVariable-Response ::= NULL	

표 3 DefineNamedVariable 프리미티브 정의

Parameter Name	req	res
	ind	con
Argument	M	
VariableName	M	
address	M	
TypeSpecification	U	
Result(+)		S
Result(-)		S
ErrorType		M

(M: mandatory, U: user Option, S: Selection)

CASM의 디바이스모델과 서버모델의 대부분은 FMS로의 매핑이 가능하나, DataObject 디바이스 모델 가운데 FC(FunctionalComponents) 객체의 일부와 서버 모델 가운데 Direct Control Service(Device Tag)는 필드버스의 응용계층 보다는 사용자계층으로 매핑되어야 한다. 이러한 매핑 방안에 대한 예로써 다음 표 4와 5에서 FC에서 MX(Measurements) 객체 중 AI(AnalogInput)와 FOUNDATION Fieldbus와 PROFIBUS의 사용자계층 간의 매핑 테이블을 나타내었다.

표 4 CASM-FOUNDATION Fieldbus 사용자계층 매핑의 예

DataObject Class	FB Class / Object
Name	
AI{	AI Block
i	OUT with SIMULATE = disabled
f	OUT with SIMULATE = disabled
q	Alarms
t)	Alert Analog Block
	TimeStamp

CASM의 AI는 아날로그 입력을 나타내며, 표 4에서 보는 바와 같이 AI의 실제 출력값을 나타내는 i, f 컴포넌트는 각각 FOUNDATION Fieldbus 사용자 계층의 FB 중 AI 블록의 OUT으로 매핑된다. 또한 오브젝트 값에 대한 유효함, 통신오류, 유효범위초과등을 나타내는 q는 FB AI 블록의 Alarms로 매핑된다. 데이터가 최

중적으로 갱신된 시각을 나타내는 t는 FB의 AI블럭에 별도의 파라미터가 없기 때문에 Alert Analog 블록의 TimeStamp 파라미터로 매핑된다. 즉, AI는 FOUNDATION Fieldbus FB의 AI 블록과 AlertAnalog 블록을 연결한 블록에 매핑된다.

표 5 CASM-FOUNDATION Fieldbus 사용자계층 매핑의 예

DataObject Class	PA Object
Name	
AI(
i	Discrete Structure Value
f	Floating Point Structure Value
q	Status with Block Object BYTE #2 = 02 BYTE #3 = 01 BYTE #4 = 01
t)	Bitstring Structure Value

CASM AI 오브젝트는 실제적인 데이터값이 정수형인지 실수형인지에 따라 PROFIBUS-PA 오브젝트의 Discrete Structure 또는 Floating Point Structure로 매핑되고, q는 해당 Discrete 또는 Floating Point Structure의 Status에 매핑된다. 이때 Block Object는 BYTE #3 = 01(Input), BYTE #4 = 01(Analog Input)으로 설정된다. 또한 TimeStamp를 나타내기 위한 t는 추가적인 Bitstring Structure를 정의하여 그 Bitstring Structure로 매핑된다.

표 6 AI 객체 매핑 의사 코드

```

DefineDataStructure():
While( !DataReceived() )
  WaitForPDU():
  CheckHeader():
  CheckService():
  /* UCA에서 필드버스로 변환하는 경우 */
  if ( SenderNetwork == UCA ) {
    /* 오브젝트 매핑 */
    if ( UCAObjType == "AI" ) {
      if ( UCAObjDataType == "INTEGER" )
        FieldbusObj.AIBlock.OUT = UCAObj.AI.i:
      else if ( UCAObjDataType == "FLOAT" )
        FieldbusObj.AIBlock.OUT = UCAObj.AI.f:
      /* AI 오브젝트가 아닌 경우 다른 과정 수행 */
    } else {
      Results = ProcessData():
      ReturntoSender(Results): }
    FieldbusObj.AIBlock.Alarms = UCAObj.AI.q:
    FieldbusObj.AlertAnalogBlock.TimeStamp=UCAObj.AI.t: }

    /* AI block의 SIMULATE 파라미터를 enable로 설정 */
    FieldbusObj.AIBlock.SIMULATE = enable:
    /* 매핑을 끝낸 데이터를 필드버스 측으로 송신 */
    CreateFieldbusPDU( FieldbusObj ):
    Results = SendPDU( "FIELDBUS", FieldbusObj ):
  }
  /* 처리결과 송신 */
  ReturntoSender(Results):
  
```

표 6에는 UCA의 CASM과 필드버스의 사용자계층간 AI 객체 매핑에 대한 의사(pseudo) 코드를 통해 실제적으로 UCA-필드버스 게이트웨이가 실제로 구현될 수 있음을 보였다.

3. 결 론

본 논문에서는 발전설비 통합 자동화통신망 구축을 위하여 가장 먼저 해결하여야 할 사안인 UCA와 필드버스의 인터페이스 방안을 제시하였다. UCA와 필드버스를 통합하는 경우 사용자는 UCA 환경에서 응용프로세스를 구현하며, 이는 필드버스의 응용계층과 사용자계층 프로토콜에 인터페이스 되어야 한다. 본 연구를 통하여 제시한 UCA-필드버스 인터페이스 방안은 사용자가 하부계층의 통신망 프로파일에 관계없이 UCA 사용자계층에서 제공하는 서비스만을 이용하여 응용 프로세스를 구현할 수 있는 환경을 제공한다.

본 논문에서는 다양한 UCA의 서비스와 오브젝트를 FOUNDATION Fieldbus와 PROFIBUS의 사용자계층과 응용계층으로 매핑시키는데 있어 대표적인 예를 들어 매핑에 일반적인 방법을 기술하고, 예로든 객체 및 서비스 매핑에 대한 의사코드를 나타내어 실제적으로 UCA-필드버스 게이트웨이를 구현할 수 있음을 보였다. 본 논문에서는 이러한 UCA-필드버스 게이트웨이를 이용한 전력설비 자동화 통신망의 구축에 대한 기본 방안만을 제시하였으며, 추후 실제로 UCA와 필드버스를 접속시킨 게이트웨이를 개발하고 이를 이용한 전력설비 자동화 통신망을 구축하기 위한 후속 연구가 수행되어야 할 것이다.

(참 고 문 헌)

- [1] Introduction to UCA version 2.0
- [2] UCA Profile Specification, Version 2.0
- [3] UCA Common Application Service Models(CASM) and Mapping to MMS
- [4] Generic Object Models for Substation and Feeder Equipment(GOMSFE)
- [5] IEC 870-6-503: TASE.2 Service and Protocol, Version 1996-08
- [6] IEC 870-6-802: TASE.2 Object Models, Version 1996-08
- [7] IEC 870-6-702: TASE.2 Application Profile, Version 1996-08
- [8] DIN 19245 PROFIBUS Standard Part 2 PROFIBUS-PA Protocol Specification Version : 1.0, 1.2.1995
- [9] PROFIBUS-PA Profile for Process Control Devices Class B
- [10] FOUNDATION™ Communication Specification Release 1.0
- [11] FOUNDATION™ Function Block Application Process: Part 1 and 2 Release