

반도체 장비의 메시지 통합을 위한 소프트웨어 구조 설계

임용목*, 황인수*, 김우성**, 박근덕***

Design of Software Architecture for Integrating of Messages between Semiconductor Equipments

Lim Yong Muk*, Hwang In Su*, Kim Woo Sung**, Park Geun Duk***

요약

반도체 제조 과정에서 생산과 관련된 모든 현상 데이터를 수집하는 것은 매우 중요한 작업이다. 수집된 자료의 분석을 통해 장비의 가동률, 고장 진단, 공정 제어 및 예측되는 장애 요소 제거 등에 활용할 수 있으며, 이는 궁극적으로 생산 효율 향상에 기여할 수 있기 때문이다. 많은 장비제조업체들이 이러한 목적으로 EES (Equipment Engineering System)을 도입했으며, 최근에는 웹 사용이 일상화되면서 HTTP/SOAP 프로토콜을 활용하여 장비 모니터링의 범위를 확대하는 방안이 제안되고 있다. 이러한 웹 기반 EES를 실현하기 위해서는 여러 반도체 장비에서 생성되는 다양한 메시지의 형식을 통합, 표준화함으로써 EDA(Equipment Data Acquisition)를 용이하게 하는 작업이 선행되어야 한다. 본 논문에서는 반도체 장비 간 통신에 사용되는 다양한 프로토콜의 분석을 통하여 상이한 형식의 정보를 통합하기 위한 방안을 제안하고 이를 지원하기 위한 소프트웨어 구조를 설계한다.

Abstract

It is very important to collect all production-related status values during the manufacturing process of semiconductor. The analysis results of the collected data can be used for the operation rate, fault diagnosis, process control and removal of predicted obstacles of equipments, eventually contributing to the improvement of production efficiency. For this purpose, many IC makers have adopted EES(Equipment Engineering System). As the use of web has become a daily life activity lately, it has been suggested to expand the scope of monitoring equipments using HTTP or SOAP protocols. To fulfill the web-based EES, EDA(Equipment Data Acquisition) should be facilitated first by integrating and standardizing various forms of messages generated from many different semiconductor equipments. In this paper, a method for integration between different types of information is suggested based on the analysis of various protocols used for the communication between semiconductor equipments. In addition, a software architecture to support the method is designed.

▶ Keyword : 반도체 장비(Semiconductor Equipment), 인터페이스 키트(Interface Kit), 통합 메시지(Integrated Message), XML

• 제1저자 : 임용목 • 교신저자 : 박근덕

• 접수일 : 2007.5.14, 심사일 : 2007.5.16, 심사완료일 : 2007. 5.24.

* 호서대학교 컴퓨터공학부 박사과정 ** 호서대학교 컴퓨터공학부 교수

*** 호서대학교 컴퓨터공학부 전임강사

1. 서론

반도체 제조 과정에서 생산과 관련된 모든 현상 데이터를 수집하는 것은 매우 중요한 작업이다. 수집된 자료의 분석을 통해 장비의 가동률, 고장 진단, 공정 제어 및 예측되는 장애 요소 제거 등에 활용할 수 있으며, 이는 궁극적으로 생산 효율 향상에 기여할 수 있기 때문이다.

반도체 산업은 장치 집약적 산업이어서 제조 공정에 매우 많은 종류의 장비가 사용된다. 이러한 다양한 장비에서 생성되는 현상 데이터를 효율적으로 수집하기 위해서는 장비와 컴퓨터간의 표준화된 통신 규약이 필수적이다. 이와 같은 요구에 따라 SEMI(Semiconductor Equipment and Materials International)에서는 반도체 장비와 호스트 컴퓨터 간의 인터페이스를 위한 데이터 통신 표준 규약인 SECS(Semiconductor Equipment Communications Standard) 프로토콜을 제정하였으며, 2000년에는 HTTP/SOAP 프로토콜을 통해 장비의 정보를 획득하기 위한 Interface A 표준을 발표하였다[1][2].

SECS 프로토콜이란 반도체 장비 간 혹은 장비와 연결되어 있는 호스트 간에 주고 받는 표준 통신 규약이며, 이는 메시지 전송을 위한 규약인 SECS-I(SEMI Equipment Communication Standard-I)과 HSMS(High-Speed SECS Message Services), 그리고 실제 통신되는 메시지 형식에 대한 규약인 SECS-II(SEMI Equipment Communication Standard-II)로 구성된다[3][4]. HSMS는 SECS-I의 단점인 느린 통신 속도와 하드웨어에 의존적인 특성을 보완하기 위해 1995년에 제안되었으며, TCP/IP를 지원함으로써 하나의 네트워크 케이블에 여러 개의 HSMS 연결을 지원할 수 있다[5].

그러나 실제 제조 환경에서는 메시지 전송을 위한 규약으로 SECS-I과 HSMS가 혼재되어 사용되고 있다. SECS-II의 메시지 형식을 따르는 장비라고 하더라도 장비 제조사마다 메시지의 형식은 상이하게 정의되어 동일한 모니터링이 어려우며 SECS-II 메시지 형식은 웹에서 사용되는 표준 데이터 형식과 차이점이 많아 웹을 통한 모니터링 시스템 구축에도 장애 요인이 된다[6].

이러한 문제를 개선하기 위하여 SEMI에서 Interface A 표준을 제정하였으나, 현재 가동되고 있는 대부분의 장비들은 Interface A를 지원하지 않기 때문에 이를 활용한 반도체 장비 모니터링은 더 많은 시간이 필요하다.

본 논문은 이러한 문제점을 극복하기 위해 [그림 1]과

같이 장비와 모니터링 시스템 사이에 메시지 형식을 통일하는 인터페이스 키트(Interface Kit)을 두고, 이 인터페이스 키트에서 수행되는 소프트웨어의 구조를 제안한다. 인터페이스 키트에서는 SECS-I과 HSMS에서 SECS-II 메시지를 추출하고, 추출된 SECS-II 메시지를 하나의 통합 메시지로 표준화한다. 이 표준화된 메시지는 웹을 이용한 모니터링이 가능하도록 XML 형식으로 정의된다.

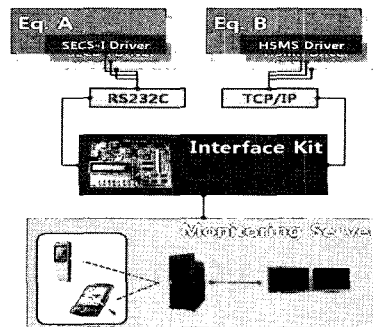


그림 1. 인터페이스 키트의 역할
Fig 1. Role of Interface Kit

II. 관련 기술 및 연구

2.1 SECS-I

SECS-I은 메시지 전송을 위한 규약이다. 이는 하나의 시작 비트와 정지 비트를 갖는 8비트 RS-232C 프로토콜을 사용하는 시리얼 통신이며, 비동기식이고 양방향통신이다. 하나의 완전한 전송 단위를 메시지라 하며 헤더의 블록 번호가 15비트이기 때문에 1 ~ 32,767 블록으로 구성된다. 데이터는 최대 254바이트의 블록 단위로 전송된다. SECS-I은 최대 9600 baud의 느린 전송 속도, 일대일 통신, 및 장비 의존성 등의 단점이 있다.

2.2 HSMS

SECS-I은 RS-232C 시리얼 통신을 사용하기 때문에, 데이터 전송 속도와 거리에 제약을 받는다. 이러한 단점을 극복하기 위해 제안된 메시지 전송 프로토콜인 HSMS는 TCP/IP를 사용하여 통신 속도 및 케이블 길이에 제한을 완화하였다. TCP/IP 프로토콜을 사용하기 때문에 독립적인 개발자들이 특정 장비에 대한 전문 지식이 없이도 기기들을 연결하고 상호 동작될 수 있다.

2.3 SECS-II

SECS-II는 반도체 제조공정에서 장비와 호스트 간에 사용되는 메시지의 구조와 기능을 규정한다. SECS-II에서는 장비와 호스트 간에 주고받는 메시지 이름이 스트림(Stream)과 함수(Function)의 조합으로 표시된다. 스트림은 메시지에 대한 분류이고, 함수는 스트림에서 특정 행동을 나타내는 메시지이다. SECS-II에서 사용하는 모든 함수는 일치하는 주와 부메시지 쌍의 순차적인 규약을 따른다. [표 1]은 SECS-II 메시지의 각 스트림 값에 따른 메시지의 의미를 요약한 것이다.

표 1. SECS-II 메시지의 의미
Table 1. Meaning of SECS-II Messages

Stream 값	메시지 의미
Stream 1	Equipment Status
Stream 2	Equipment Control and Diagnostics
Stream 3	Material Status
Stream 4	Material Control
Stream 5	Exception Reporting
Stream 6	Data Collection
Stream 7	Process Program Management
Stream 8	Control Program Transfer
Stream 9	System Errors
Stream 10	Terminal Service
Stream 11	Host File Services (1989년에 삭제)
Stream 12	Wafer Mapping
Stream 13	Unformatted Data Set Transfers

SECS-II 규약이 SEMI의 표준 규약이지만, 각 장비의 특성에 맞게 새로운 메시지를 사용자가 정의할 수 있도록 하였으며, 이러한 사용자 정의 메시지는 SECS-II 메시지의 의미를 일관되게 파악하는데 장애 요소가 된다.

2.4 Interface A

Interface A는 반도체 제조장비 제조업체들의 소프트웨어 어플리케이션의 데이터 수집과 공장 장비간의 통신을 용이하게 하기 위한 개선된 방법이다. 이 표준은 HTTP에 대한 SOAP/XML 메시지를 사용하는 EDA나 HTTP 연결에 대한 표준화된 인터페이스를 제공한다. Interface A SEMI 표준은 E120, E125, E132와 E134를 포함한다.

Interface A는 최근에 제안된 표준으로 현재 가동되고 있는 대부분의 장비들은 Interface A를 지원하지 않기 때문에, Interface A가 목표로 하는 반도체 장비에 대한 표준

화된 모니터링이나 웹을 통한 모니터링 등에 적용하기에는 아직까지는 많은 제약을 갖는다.

2.5 SML

GW Associate(現 Asyst)사의 SML(SECS Message Language)은 SEMI E5 표준의 SECS-II 메시지 표시법을 사용자가 이해하기 쉬운 태그로 표현한 것이다[7]. SML은 데이터 형식(목록, 정수형, 실수형, 문자형, 데이터 등)에 따라 각각을 태그로 정의하고, 이를 SEMI E5 표준의 표기법과 유사한 형태의 표기법을 제시함으로써 사용자의 이해를 높이고 사용의 편의성을 제공하였다. SML은 스트림과 함수 코드, 대기 비트, 리스트, 아이템 등으로 구성되어 있다. 아이탬은 타입, 개수, 값으로 나타낼 수 있으며, 아이탬의 타입은 간단한 기호로 표현된다.

그림 2. S5F1의 SML 표현
Fig 2. S5F1 Representation in SML

[그림 2]는 S5F1 메시지를 SML로 표현한 것이다. 한 개의 리스트와 세 개의 아이탬으로 구성되어 있고, 아이탬은 각각 Boolean, 4Byte unsigned Integer, ASCII의 자료형의 값을 취한다.

SML은 SECS-II 메시지를 사용자가 이해하기 쉬운 형태로 표현하고 있지만, XML 형식이 아니기 때문에, 웹을 통한 모니터링에 활용하기에는 적합하지 않다.

2.6 XML 메시지 형식

XML(eXtensible Markup Language)은 웹 상에서 구조화된 문서를 전송 가능하도록 설계된 마크업 언어(Markup language)이다[8]. 고정된 요소(Tag, Element)만을 사용한 HTML과는 달리 자유롭게 문서의 요소와 속성, 개체를 선언하여 자료를 구조적으로 표현할 수 있으며, 스타일시트(XSL)에 의해 다양한 형태의 문서를 나타낼 수 있다.

웹 문서의 형식을 표준화된 구조로 표현하기 위해 제안된 XML은 메시지 형식의 통일이 필요한 반도체 장비 분야에 적용하기 적합하며, XML로 표현된 정보는 웹 기반 모니터링 구축을 용이하게 하는 장점이 있다. [9]에서는 반도체 장비간 통신을 위한 XML데이터 구조를 제안하였다. 이러한 XML 구조를 따르는 메시지를 얻기 위해서는 장비와

모니터링 시스템 사이에 메시지 형식을 통일하는 소프트웨어의 지원이 필요하다.

III. 소프트웨어 구조 설계

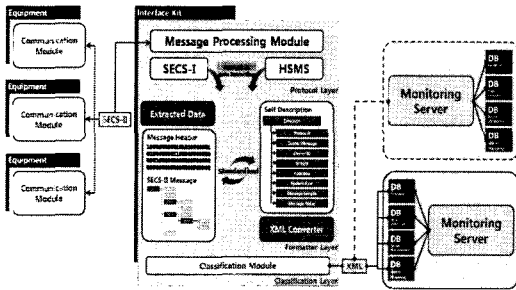


그림 3. 메시지 통합을 통한 소프트웨어 구조
Fig 3. A Software Architecture for Integration of Messages

본 논문에서 제안하는 메시지 통합을 위한 소프트웨어 구조는 [그림 3]에 제시되어 있다. 인터페이스 킷에서는 반도체 장비와의 통신을 위한 프로토콜인 SECS-I과 HSMS를 모두 지원하며 각 프로토콜에서 필요한 데이터를 추출한다. 추출된 데이터는 자기기술(Self Description)에 맞게 변환되어 XML 형태로 규격화된다. 완성된 XML 메시지는 분류 모듈(Classification Module)을 통하여 장비 모니터링 소프트웨어에서 활용된다.

[그림 4]는 인터페이스 킷의 기능을 소프트웨어 측면에서 계층화한 것이다.

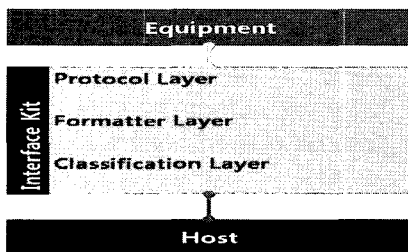


그림 4. 인터페이스 킷의 계층도
Fig 4. Layers of Interface Kit

인터페이스 킷은 시리얼 또는 병렬 접속을 통해 적어도 하나 이상의 반도체 설비장비와 연결된다. 여기서 반도체 설비장비라 함은 반도체 제조 공정에 필요한 일반적인 장비들을 의미

하는 것으로, 세정장비, 식각장비, 도포장비, 포토(Photo)장비, CMP(Cheical and Mechanical Polishing) 장비 뿐 아니라 제품 검사장비 등을 포함한다. 이러한 설비장비들은 각각의 공정을 수행하기 위한 제어기를 포함하거나 장비 내의 상태를 체크할 수 있는 여러 형태의 센서 장비들을 포함할 수 있다.

인터페이스 킷은 서버와 연결되어 반도체 설비장비로부터 전달받은 메시지를 XML 형태로 변환하여 규격화한 후, 서버로 전송하는 역할을 담당한다. 인터페이스 킷은 SECS-I 및 HSMS의 헤더를 분석하여 규격화에 필요한 데이터를 추출하는 프로토콜 계층(Protocol Layer), 추출한 데이터와 SECS-II 메시지를 XML로 규격화하기 위한 규격화 계층(Formatter Layer), 변환된 XML 메시지를 분류하기 위한 분류 계층(Classification Layer)으로 구성된다.

여러 반도체 장비에서 생성되는 수많은 메시지는 그 성격에 따라 분류하고, 모니터링 목적에 맞는 데이터베이스에 저장되어, 분산 모니터링에 활용되어야 한다. 이러한 목적을 충족하기 위해 분류 계층이 필요하며, XML 형식으로 표현된 메시지에도 분류에 대한 정보가 포함되어야 한다.

3.1 프로토콜 계층

SECS-I의 시리얼 통신과 HSMS의 TCP/IP 통신은 서로 다른 형태의 헤더를 갖는다. SECS-I 프로토콜은 메시지를 블록단위로 전송하기 때문에 그에 따른 처리가 필요하다. 또한 HSMS의 경우 데이터 메시지 이외에 통신에 관련된 컨트롤 메시지가 존재한다. 프로토콜 계층은 이렇게 서로 다른 형태의 프로토콜을 분석하고 규격화에 필요한 데이터를 추출하거나 관리하는 계층이다. 이 프로토콜 계층은 SECS-I 처리 모듈과 HSMS 처리 모듈로 나뉜다.

3.1.1 SECS-I 처리 모듈

SECS-I의 가장 큰 특징은 RS-232C 즉, 시리얼 통신을 사용하기 때문에 블록 단위의 전송을 해야 한다. 최대 254 바이트까지 전송할 수 있지만 메시지가 최대 바이트를 초과하게 되면 블록 단위 전송이 이루어지게 된다. 한 메시지를 다 수신하게 되면 헤더를 분석하여 규격화에 필요한 데이터를 추출하게 된다.

표 2. 블록 헤더 구조
Table 2. Structure of a Block Header

	8	7	6	5	4	3	2	1
1	R	상위 Device ID						
2	하위 Device ID							
3	W	Stream						
4	Function							
5	E	상위 Block ID						
6	하위 Block ID							
7	시스템 Byte							
8	시스템 Byte							
9	시스템 Byte							
10	시스템 Byte							

[표 2]는 SECS-I의 헤더 구조에 대한 표이다. R은 역순 비트로서 메시지의 방향을 나타낸다. 장비로 가는 메시지는 0으로 호스트로 가는 메시지는 1로 설정된다. 그 다음의 W는 대기 비트로 주 메시지의 송신자가 응답을 기대함을 나타낸다. 대기 비트 값이 1이면 응답을 기대된다는 것을 의미하고 0이면 그 반대가 된다. 마지막으로 E는 종료 비트로서 블록이 마지막 블록임을 정의한다. 종료 비트의 값이 1이면 그 블록이 마지막 블록임을 의미하고, 0이면 뒤 이어 다른 블록이 있음을 의미한다. 이 종료 비트를 검사하여 마지막 블록까지 전송 받게 되면 필요 데이터를 추출하여 저장하게 된다.

3.1.2 HSMS 처리 모듈

HSMS 메시지는 데이터 메시지와 컨트롤 메시지로 나누어진다. 여기서 데이터 메시지는 데이터를 전송하는 메시지가 되고, 컨트롤 메시지는 통신 관련 메시지가 된다. 통신 관련 메시지는 초기 연결 설정, 연결 확인 혹은 연결을 해제하는 등의 메시지를 말한다.

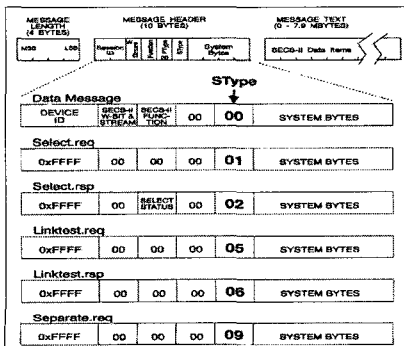


그림 5. HSMS 메시지 형태
Fig 5. HSMS Message Format

[그림 5]에서 보는 바와 같이 각 메시지들은 S타입에 의해 구분된다. 그렇기 때문에 HSMS 처리 모듈에서는 메시지를 수신하게 되면 S타입을 검사하여 컨트롤 메시지일 경우와 데이터 메시지일 경우를 분리해서 처리한다.

데이터 메시지일 경우엔 장비 ID, 대기 비트, 스트림, 펄스 등 필요한 데이터를 추출하여 저장하고, 컨트롤 메시지일 경우 데이터를 추출하여 다른 루틴을 통하여 데이터 메시지와는 별도로 처리한다.

3.2 규격화 계층

규격화는 모든 장비의 제어메시지/데이터에 대응하는 하나의 통합된 메시지를 작성하는데 목적이 있다. 통합된 메시지는 XML형태의 메시지로써 장비에서 나오는 신호의 합집합이라 할 수 있으며, 각종 장비에서 나오는 수많은 메시지/데이터에 맞는 하나의 규격을 디자인 하고 규격에 맞게 기존 메시지를 변환 시키는 역할을 한다.

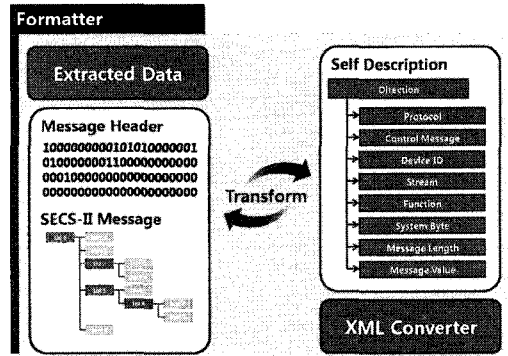


그림 6. 규격화 계층의 구성
Fig 6. Structure of Formatter Layer

[그림 6]에서는 규격화 계층의 구성을 보여주고 있다. 규격화 계층은 추출 데이터 처리, 자기기술 맵핑, XML 변환기로 이루어져 있다.

표 3. 자기기술의 각 요소 의미
Table 3. Meaning of Self Descriptions

요소	내 용
Direction	메시지의 특성에 따라서 호스트를 다르게 관리하기 때문에 수신한 메시지를 담당하는 호스트로 전송되어야 한다.
Protocol	수신된 메시지가 사용하는 통신 프로토콜의 종류를 기록 (SECS-I, HSMS)

Control Msg.	(HSMS만 해당) 컨트롤 메시지를 전송한 장비의 내용 기록
Device ID	메시지를 전송한 장비의 ID
Stream	메시지의 스트림
Fuction	메시지의 평선
SystemByte	메시지들의 구별을 위해서 필요한 바이트
Message Length	메시지의 크기
Message Value	SECS-II 메시지

장비에서 전송된 데이터는 추출데이터 처리 모듈에 의해 처리/분석되고 규격화에 필요한 데이터를 추출한다. 이 데이터들을 [표 3]의 자기기술에 맞게 맵핑되며, 다시 XML 변환기에 의하여 통합된 XML 메시지로 변환 시킨다. XML 변환기에서는 자기기술 요소와 맵핑된 헤더부, 메시지를 표현하는 표현부로 구분하여 통합된 XML 메시지를 완성한다

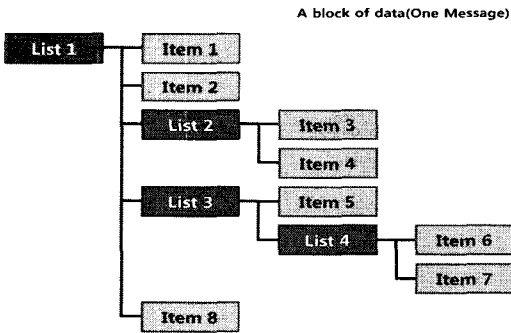


그림 7. SECS-II 메시지의 구조
Fig 7. Structure of SECS-II Message

헤더부는 맵핑을 통해 만들어진 데이터를 이용하고, 표현부는 (그림 7)과 같이 SECS-II 메시지의 트리를 표현하도록 XML 스키마를 작성한다. 작성된 스키마는 [표 4]와 같이 리스트를 재귀적인 형식으로 정의하였다.

표 4. SECS-II 메시지의 XML 스키마
Table 4. XML Schema for SECS-II Messages

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="Item">
    <xsd:complexType>
```

```
<xsd:sequence>
  <xsd:element name="Type" type="xsd:string"/>
  <xsd:element name="Value" type="xsd:string"/>
</xsd:sequence>
<xsd:sequence>
  <xsd:attribute name="Sequence"/>
</xsd:complexType>
</xsd:element>
<xsd:element name="List">
  <xsd:complexType>
    <xsd:choice maxOccurs="unbounded">
      <xsd:element ref="List"/>
      <xsd:element ref="Item"/>
    </xsd:choice>
    <xsd:attribute name="Count"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="Message">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="Header">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="Direction" type="xsd:string"/>
            <xsd:element name="Protocol" type="xsd:string"/>
            <xsd:element name="ControlMsg" type="xsd:string"/>
            <xsd:element name="DeviceID" type="xsd:string"/>
            <xsd:element name="Stream" type="xsd:integer"/>
            <xsd:element name="Function" type="xsd:integer"/>
            <xsd:element name="SystemByte" type="xsd:integer"/>
            <xsd:element name="DataLength" type="xsd:integer"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="MessageValue">
        <xsd:complexType>
          <xsd:choice>
            <xsd:element ref="List"/>
            <xsd:element ref="Item"/>
          </xsd:choice>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
</xsd:schema>
```

스키마를 통해 SECS-II 메시지를 XML 형식으로 변환하면 메시지는 [표 5]와 같은 형태를 갖는다.

[표 5]는 SECS-II 메시지 중 S5F1을 예로 작성된 것으로 헤더부는 <Header>로 메시지 표현부는 <MessageValue>로 구분되어 있다. 'List' 요소에 'Count'라는 속성을 사용해서 해당 리스트에 포함된 아이템의 수를 표현하며, 'List' 요소는 또 다른 하위 'List' 요소를 포함 할 수 있다.

표 5. 통합된 XML 메시지의 예
Table 5. Example of an Integrated XML message

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Message xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="s5f1.xsd">
```

```

<Header>
  <Direction>Exception</Direction>
  <Protocol>HSMS</Protocol>
  <ControlMsg> </ControlMsg>
  <DeviceID>EW12003</DeviceID>
  <Stream>5</Stream>
  <Function>1</Function>
  <SystemByte>210</SystemByte>
  <DataLength>30</DataLength>
</Header>
<MessageValue>
  <List>
    <Count>3</Count>
    <Item>
      <Type>Binary</Type>
      <Value>0100</Value>
    </Item>
    <Item>
      <Type>Integer</Type>
      <Value>17</Value>
    </Item>
    <Item>
      <Type>ASCII</Type>
      <Value>T1 HIGH</Value>
    </Item>
  </List>
</MessageValue>
</Message>
    
```

3.3 분류 계층

본 논문에서 제안한 XML 구조는 향후 분산 처리가 가능하도록 분류를 위한 정보를 포함하고 있다. 분류 계층은 메시지 분류가 가능하도록 [표 3]에서의 <Direction> 요소를 이용하여 메시지를 분산 및 수집하는 계층이다.

표 6. 메시지 분류
Table 6. Category for Messages

범주	메시지 종류
예외 (Exception)	S5F1, S5F3, S5F9, S5F13 등
데이터 수집 (Data Collection)	S6F9, S6F11, S6F13 등
레시피 관리 (Recipe Management)	S15F1, S15F3, S15F5, S15F7 등
웨이퍼 맵핑 (Wafer Mapping)	S12F1, S12F3, S12F5, S12F7, S12F9 등

SECS-II 메시지는 메시지의 성격에 따라 예외, 데이터 수집, 레시피 관리 및 웨이퍼 맵핑 등의 범주로 분류할 수 있다. SECS-II 메시지의 분류는 [표 6]에 요약되어 있다. XML 형식으로 표현된 메시지는 <Direction> 요소에 근거하여 특성에 맞는 호스트로 분산 전송되어 저장될 수 있다.

3.4 프로그램 프로토타입

본 논문에서 제안한 소프트웨어 구조에 따라 각 소프트웨어 컴포넌트의 프로토타입을 구현하여 XML 형식으로 통합된 메시지 생성을 테스트하였다. [그림 8]은 구현한 프로토타입의 실행 화면을 보여준다.

테스트 환경은 실제 장비 대신 시뮬레이터를 사용하였으며, 인터페이스 킷은 PC를 활용하였다. 구현된 시스템은 [표 7]과 같이 모듈과 테스트 프로그램으로 구분된다. 테스트에서는 시뮬레이터에서 나온 메시지를 본 논문에서 제시한 XML 스키마에 맞게 XML 형식으로 메시지가 생성되는지 확인하며, 이 메시지를 이용하여 분산 모니터링이 가능한지를 검증한다.

표 7. 컴포넌트 요약
Table 7. Summary for Components

구분	명칭	내용
A	시뮬레이터	장비 기능을 하는 상용 시뮬레이터
B	메시지 처리 모듈 테스트 프로그램	메시지 처리 모듈을 테스트하는 프로그램
C	규격화 모듈	메시지 처리 모듈에서 추출한 데이터를 자기기술요소에 맵핑하는 프로그램
D	예외 서버	경보관리 서버 데모 프로그램
E	데이터 수집 서버	데이터 수집 서버 데모 프로그램
F	XML 변환기	XML 변환기를 통해 통합된 XML 메시지

테스트 결과 SECS-I과 HSMS 프로토콜을 사용하는 메시지 모두 추출 가능하였고, 모두 XML 스키마에 준하는 메시지 형태로 변환되었다.

IV. 결론 및 향후 과제

본 논문에서는 다양한 반도체 장비에서 생성되는 메시지를 효율적으로 모니터링하기 위하여 각 장비에서 사용되는 메시지의 형식을 통합하기 위해 XML 스키마를 제시하였고, 제시한 XML 스키마를 따르는 XML 문서를 산출하기 위한 소프트웨어 구조를 제안하였다.

메시지를 XML 형식으로 통합하여 표현함으로써 일관된 방식으로 장비 관리가 가능하며, 웹을 이용한 모니터링에 활용할 수 있다. 또한 장비 집약적인 반도체 제조 설비에서의 수많은 메시지로 인한 부하를 분산 관리하는데 용이한 장점이 있다.

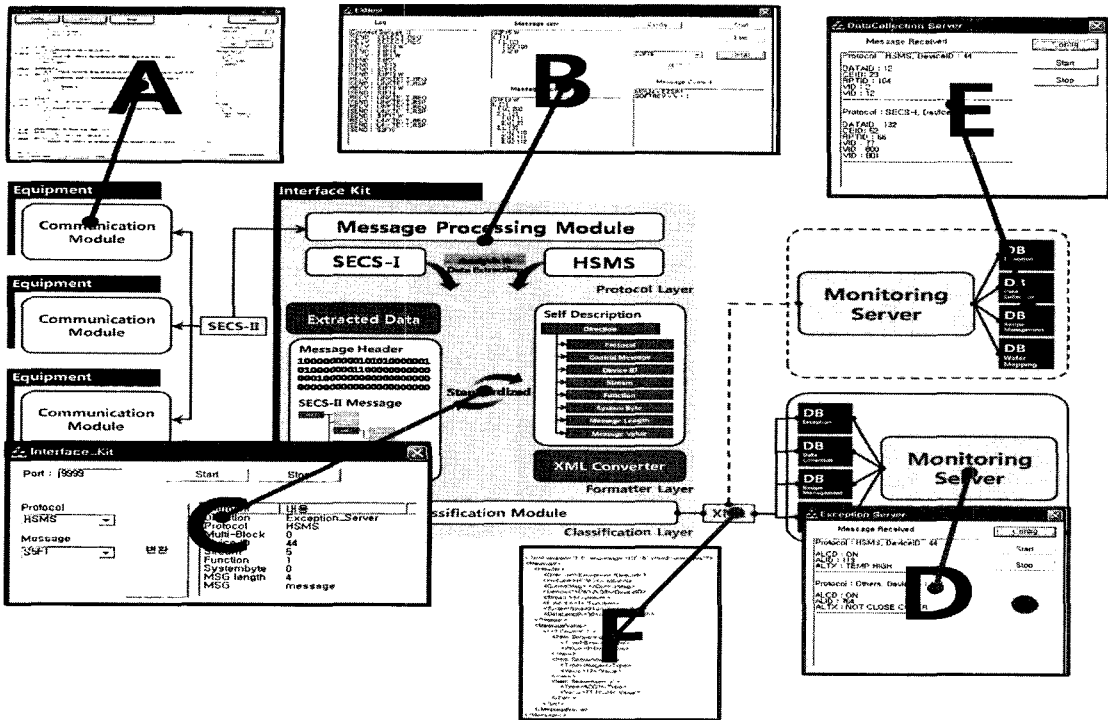


그림 8. 프로그램 실행 화면
Fig 8. Screen Shots for Demo Program

현재는 시뮬레이터를 이용하여 테스트하였지만, 향후 현재 제작 중인 인터페이스 킷 장비에 본 논문에서 제안한 소프트웨어를 설치하여 실제 환경에서 유용성을 검증하고, XML 메시지 생성 성능을 분석할 예정이다. Interface A 표준은 현재 적용에 어려움이 있지만 반도체 제조장비 및 설비에서 차세대 산업 표준으로 인식되고 있다. 따라서 향후 제안한 인터페이스 킷과 Interface A의 호환에 대한 연구를 진행할 것이다.

참고문헌

[1] <http://www.semi.org>

[2] SEMI, "SEMI E120-0706 Common Equipment Model Standard", www.semi.org

[3] SEMI, "SEMI E4-0699 SEMI Equipment Communications Standard 1 Message Transfer (SECS-I)", www.semi.org

[4] SEMI, "SEMI E5-0706 SEMI Equipment Communications Standard 2 Message Content (SECS-II)", www.semi.org

[5] SEMI, "SEMI E37-0303 High-Speed SECS Message Services (HSMS) Generic Services", www.semi.org

[6] Fan-Tien Cheng, Meng-Tsang Lin, Rong-Shean Lee, "Developing a Web-enabled Equipment Driver for Semiconductor Equipment communications", Proceedings of the 2000 IEEE International conference on Robotics & Automation, April 2000

[7] GW Associates, Inc., "SML(SECS Message Language)", <http://www.gwa.com/secsinfo/secsinfo.htm>

[8] Bray, Tim, et. al., "Extensible Markup Language (XML) 1.0," W3C Recommendation, February 10,

1998(<http://www.w3.org/TR/1998/REC-xml-19980210>)

- [9] 반도체 장비 간 통신을 위한 XML데이터 구조에 관한 연구, 황민정, 인하대학교 대학원, 2002

저자 소개



임용묵

2004년 2월 호서대학교 컴퓨터공학과 공학학사

2006년 2월 호서대학교 컴퓨터공학과 공학석사

2006년 3월 ~ 현재 : 호서대학교 컴퓨터공학과 박사과정



황인수

1993년 2월 호서대학교 전자계산학과 공학학사

1995년 2월 호서대학교 전자계산학과 공학석사

2004년 3월 ~ 현재 : 호서대학교 컴퓨터공학과 박사과정

2001년 12월 ~ 현재 : (주)에이 스택 대표



김우성

1993년 2월 서강대학교 대학원 컴퓨터공학과 (공학박사)

1987년 2월 ~ 현재 호서대학교 컴퓨터공학부 교수

관심분야 : 지능정보 시스템, 장비 제어, XML 응용



박근덕

2005년 8월 : 서울대학교 전기컴퓨터공학부 (공학박사)

2006년 3월 ~ 현재 : 호서대학교 컴퓨터공학부 전임강사

관심분야 : 웹공학, 서비스 지향 컴퓨팅, XML 응용