

URC 로봇과 IT 기술의 컨버전스를 위한 메시지 프로토콜 설계

Design of a Message Protocol for The Convergence of a URC-Robot and Information Technologies

임용묵, 김우성, 박성준*
호서대학교

Lim Yong-Muk, Kim Woo-Sung, Park Sung-Jun*
Dept of Computer Engineering, Hoseo University,
*Dept of Game Engineering, Hoseo University

요약

최근 로봇 시스템 보급과 실용화를 앞당기기 위한 URC 로봇에 대한 연구가 활발해 지면서 로봇 플랫폼 기술개발 및 표준화에 대한 관심이 높아지고 있다. 하지만 현재의 URC 로봇의 표준화 연구는 과거 장비 및 반도체 장비에서의 표준화 연구와 같은 오류를 반복하고 있다. OPC에서는 PLC 장비를 위하여 OPC규약을 만들었고 SEMI에서는 반도체 장비를 위해 SECS-II 규약을 만들게 되었다. 이는 장비가 많고 다양해지면서 이 기종의 장비들의 메시지를 통합환경에서 제어 및 모니터링 하는데 문제가 발생하였기 때문이다. 하지만 URC 로봇의 플랫폼 개발에 있어 메시지 규약에 관한 사항은 배제되어 있다. 특히 URC 로봇은 유비쿼터스 환경의 핵심인 IT 기술과의 컨버전스가 이루어 져야 하기 때문에 규격화된 메시지 프로토콜의 필요성은 더욱 절실하다. 따라서 본 논문에서는 URC 로봇 플랫폼 개발에 있어 규격화된 메시지의 필요성을 제시하고 RUPI, OPC, SECS-II 메시지를 분석하여 IT 기술과의 컨버전스에 기반이 되는 메시지 프로토콜을 설계한다.

Abstract

As research for URC robot to advance the robot system dissemination and practical use becomes active, also interest for robot platform and standardization is increasing. However, URC robot middleware does not involve a convergence with IT. Although it seems to be a high connection between robots and IT, the existing middleware has not considered different environments, thus making convergence efforts difficult. The middleware should improve their functions in order to synchronize with IT and to interact through formatted messages. In this paper, we present the necessity of the formatted message in the development of URC platform. In addition, design of the message protocol for the convergence based on the analysis of RUPI, OPC and SECS-II.

I. 서론

유비쿼터스 세상은 시간과 공간의 간격을 좁힌다는 의미로서, 서비스 사용자와 제공자는 주변의 여건에 가능한 의식하지 않아야 한다. 이와 함께 관련 기술에 대한 끊임없는 기술개발이 진행되면서 URC(Ubiquitous Robot Companion)라는 새로운 개념의 로봇이 등장하게 되었다[5]. URC 로봇은 언제 어디서나 나와 함께 하며 나에게 필요한 서비스를 제공하는 로봇으로 로봇 핵심 기능을 외부 서버에 전달시키고 단순한 하드웨어 구

성으로 다양한 서비스를 실시간으로 제공한다.

현재 URC 로봇은 정보, 홈케어, 교육, 오락, 감성 등 많은 서비스를 제공하고 있으나 기존의 유비쿼터스 환경을 제공하기 위한 핵심 기술인 IT 기술과의 컨버전스에는 어려움이 있다. 로봇과 IT 기술의 컨버전스를 위해서는 이기종의 하드웨어나 프로토콜, 통신환경 등을 연결하여, 응용프로그램과 그 프로그램이 운영되는 환경 간에 원만한 통신을 이룰 수 있는 미들웨어가 필요하다[6]. 하지만 현재의 URC 로봇의 미들웨어는 실시간 동기화 및 정형화된 메시지를 통한 상호 작용등의 기능이 포함되어 있지 않다.

OPC(OLE for Process Control)는 OPC 메시지 규약을, SEMI(Semiconductor Equipment and Materials International)는 SECS-II(SECS Equipment Communication Standard 2)메시지 규약을 제정하여 각각 PLC(Programmable Logic Controller)와 반도체 장비를 통합 관리 및 모니터링 하였다. OPC 및 SECS에서 메시지 규약을 정의 한 것은 매우 많은 종류의 장비가 있으며 많은 종류만큼 다양한 통신 규약 및 메시지가 사용되고, 제조 과정에서 생산과 관련된 모든 현상 데이터를 수집하는 것이 매우 중요하여 수집된 자료의 분석을 통해 장비의 가동률, 고장 진단, 공정 제어, 장애요소 제거 등에 활용할 수 있기 때문이다[7]. URC 로봇 또한 위와 같은 환경에서 크게 벗어나진 않는다. URC 로봇은 외부환경을 인식하고 스스로 상황을 판단하여 자율적으로 동작하는 지능형 로봇과 달리 로봇 핵심 기능을 외부 서버에 전달시킨다. 이것은 현상 데이터를 실시간으로 서버로 전송하고 수집 및 분석 할 수 있어야 한다는 것이며, 여러 종류의 로봇이 있을 수 있고 그만큼 다양한 통신 규약 및 메시지가 사용되어 진다는 것과 같다. 하지만 현재의 URC 로봇 플랫폼 개발에 있어 규격화된 메시지 프로토콜 및 유비쿼터스의 핵심인 IT 기술과의 컨버전스는 배제되어 있다.

따라서 본 논문에서는 URC 로봇과의 IT 기술과의 컨버전스를 위한 메시지 프로토콜을 설계한다. 제안하는 메시지 프로토콜은 네트워크 기반 로봇의 표준 환경을 제공하는 RUPI(Robot Unified Platform Initiative)의 메시지 구조와 OPC 및 SECS-II 메시지 구조를 분석하여 기존의 장비 뿐 아니라 URC 로봇에 이르기까지 IT 기술의 핵심인 웹서비스와 융합할 수 있도록 XML(eXtensible Markup Language)을 기반으로 하며 다양한 클라이언트와의 양방향 통신을 포함하도록 한다.

II. 관련기술 연구

1. 지능형 로봇 플랫폼

〈표 1〉은 국내·외 지능형 로봇 플랫폼기술을 비교 분석한 것이다. 국내 주요 S/W 플랫폼 기술개발 현황을 살펴보면, 한국전자통신연구원의 RUPI, 한국과학기술

연구원원의 로봇 미들웨어 플랫폼, 그리고 한국생산기술 연구원에서 퍼스널로봇에 적용가능한 S/W 플랫폼을 개발하고 있다[8]. 국내의 개발은 S/W 플랫폼간의 상호 호환 및 다양한 통신 정보기기와의 상호 운용성 등을 갖는 로봇 S/W 표준 플랫폼 및 제반 표준 규격에 활발한 연구 개발이 진행되고 있으나, 이 기종 로봇사이의 대화를 위한 규격화된 메시지 통합에 대한 연구 및 기술은 전무한 상태이다.

국외는 전반적으로 S/W 구조 및 컴포넌트 기술은 여러 가지 형태로 연구 중에 있으며, Agent 기반의 S/W 모듈로 개발하고 있다. 로봇의 핵심 알고리즘 부분은 성숙되어 있으나 미들웨어 및 컴포넌트로 wrapping 하려는 시도는 아직까진 없다. 미들웨어 기술 또한 다양한 미들웨어를 개발 중에 있으며, S/W 개발도구의 경우는 많은 프로그램이 상용화되어 있으나, 개발 도구와 연동은 MS의 Robotics Studio만 가능한 실정이다.

표 1. 국내·외 지능형 로봇 플랫폼 기술 비교

구분	외국	국내
로봇 S/W 구조 및 컴포넌트 표준기술	<ul style="list-style-type: none"> · 많은 구조에 대해 연구중 · OMG에서는 표준을 시도 중 	<ul style="list-style-type: none"> · S/W구조 및 컴포넌트에 대해 표준화에 대한 논의는 되지 않고 있음 · ETRI와 생산기술연구원등을 통해 S/W 모듈을 컴포넌트 작업을 하고 있음
코어 알고리즘 기술	<ul style="list-style-type: none"> · 매우 성숙되어 있음 · 많은 연구자들이 참여하며 다양한 알고리즘을 개발 	<ul style="list-style-type: none"> · 현재 코어 알고리즘을 선진국 수준으로 따라가기 위하여 많은 연구 중 · 로봇별로 연구 하고 있어 연구의 활용성이 낮음
로봇 미들웨어 기술	<ul style="list-style-type: none"> · 다양한 미들웨어 (CORBA 기반, 자체 개발)를 개발 · OMG에서는 표준화를 시도 중 	<ul style="list-style-type: none"> · 생산기술연구원 및 강원대학교에서 로봇에 적합한 경량형 자체 미들웨어 · ETRI에서는 CORBA를 사용하여 배치 미들웨어 · KIST에서는 UPnP를 기반으로 한 미들웨어 · 표준화에 대해 토의하고 있는 중임
로봇 S/W 개발도구 기술	<ul style="list-style-type: none"> · OMG에 참여하는 일본측에서는 개발하여 운영 중 · ER 사에서도 상품화하여 출시 · 시뮬레이션 프로그램도 많이 사용화되고 있음 · 개발도구와 시뮬레이션의 연동은 아직 되지 않음 	<ul style="list-style-type: none"> · 생산기술연구원 및 KIST에서 개발한 미들웨어를 동작시키기 위해서 소형 개발도구를 개발 · 시뮬레이션 등 다른 도구와의 연계가 미흡함

2. 메시지 분석

2.1 RUPI

RUPI은 다양한 로봇 플랫폼이 서버와 연동하여 다양한 로봇 서비스를 수행할 수 있도록 지원하는 네트워크 기반 로봇의 표준 환경을 제공하는 제반 규격 및 플랫폼이다[6]. RUPI 규격은 크게 RUPI 서버, RUPI 클라이언트, 통신 및 응용컴포넌트 인터페이스, 개발도구 등으로 구성되어 있다. 이중 통신 프로토콜 규격은 3종으로 URC 서버/클라이언트 통신 프로토콜(RPC 계층), URC 서버/클라이언트 통신 프로토콜, 썬 클라이언트 로봇을 위한 디바이스 맵 기반의 URC 서버/클라이언트 통신 프로토콜이 정의 되어 있다. 메시지 구조는 총 5종으로 연결요청(connect), 메시지 헤더(header), 함수 호출(call), 결과값 전달(reply), Error(error) 전달을 포함한다.

<그림 1>과 <그림 2>는 RUPI 메시지 헤더와 <call> 메시지 구조를 보여 주고 있다.

```
<header> ::= <int:magic> //magicnumber(970208)
           <int:request_id> // 요청식별자
           <int:length> // 메시지길이
           <byte:code> // 메시지타입
```

▶▶ 그림 1. RUPI 메시지 헤더

```
<call> ::= <header> // 메시지헤더
         <string[:path]> // 대상 servant의 path
         <type:declaring_type> // 호출 메소드가 속한 타입
         <string:method_name> // 호출 메소드 이름
         <int:arg_count> // 호출 인자 개수
         <typed:arg>*arg_count // 호출 인자 값들
```

▶▶ 그림 2. RUPI <call>메시지 구조

2.2 OPC

국내의 자동화 장비 인터페이스는 대부분이 RS-232C, RS-422/485C등의 레거시 인터페이스를 중심으로 구성이 되어 있다. 또한 정보의 중앙 집중화를 위해 필드버스 등의 환경을 구축하지만 제어 환경은 여전히 한계를 지니고 있다. 최근에는 TCP/IP를 이용한 환경 구축을 적극 도입하고 있지만 레거시 시스템과의 교체 및 호환성 문제로 쉽게 이루어지지 못하고 있는 실정이다.

이러한 문제 해결 및 새로운 기술 발전을 도모하기 위하여 자동화 장비를 위한 국제적인 표준 OPC규약이 제정되어 그와 관련한 많은 연구가 진행되고 있다.[1]

OPC 규약은 OPC 서버와 클라이언트를 제공하여, 서비스 제공자와 사용자를 구분하였다. 초기에는 OPC 서버를 자동화 장비 제조업체에서 제공하지 않고 어플리케이션 개발 업체에서 제공하였으나, 현재는 장비 제조업체에서도 자체 장비에 OPC 서버 기능을 제공하는 모듈을 탑재하는 추세이다. OPC 클라이언트는 대부분이 HMI 장비 제조사에 의해 제공되고 있다.[2]

다음 <표 2>는 OPC 컨버터의 워드단위 2블록 연속 읽기 명령 변환 예를 보여주고 있다.

레거시 인터페이스 통신과 OPC-DA 통신은 하나의 HMI 운용시스템에 모두 구현되어 있으며, 운용 시스템의 선택에 따라 두 개의 통신 방법 중에서 하나를 선택하여 통신할 수 있도록 설계되었다. 이는 기존의 레거시 통신 방식을 지원하면서 새로운 OPC 클라이언트 인터페이스를 추가함으로써 확장성과 전용 통신 모듈만이 가능한 PLC의 직접적인 제어의 장점을 혼합하기 위한 것이다. 이를 위하여 OPC 변환 모듈을 새로 작성하였으며, 워드 단위의 2블록 연속 읽기 명령의 변환 예를 나타낸 것이다.[3]

표 2. OPC 컨버터의 워드단위 2블록 연속 읽기 명령 변환 예

통신방식	명령
레거시 통신	OORSB08%QW0.1.102
OPC 통신	Item[1] 주소 : %QW0.1.1 Item[1] 자료형 : WORD Item[2] 주소 : %QW0.1.2 Item[2] 자료형 : WORD

2.3 SECS-II

GW Associate(現 Asyst)사의 SML은 SECS-II의 메시지 표시법을 사용자가 이해하기 쉬운 Tag로 표현한 것이다. 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.[4] SML은 SECS-II에서의 정의와 마찬가지로 스트림과 평선의 코드를 나타내고, 대기 비트

(Wait bit), 리스트(List), 마지막으로 아이템(Item)을 나타낸다. 아이템은 타입(Type), 개수, 값으로 나타낼 수 있으며, 아이템의 타입은 간단한 기호로 표현된다.

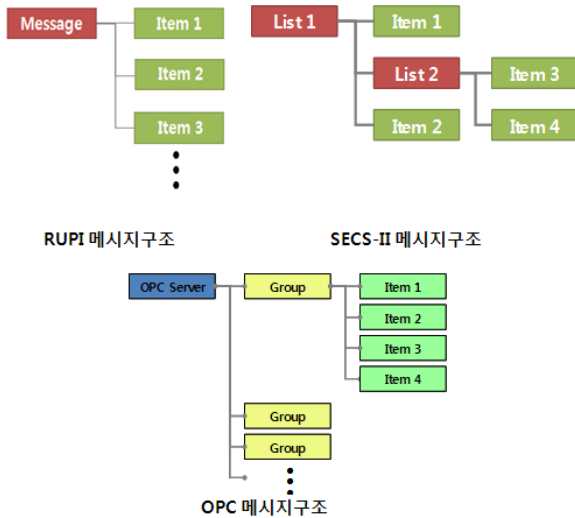
```
S5F1 W *H<-E
<L3
<B[1]ALCD> *ALCD-Alarm On/Off and Severity
<U4 ALID> *ALID-Alarm ID
<A[40]ALT< > *ALT<-Alarm Text
>
```

▶▶ 그림 3. S5F1의 SML 표현

〈그림 3〉은 S5F1 메시지를 SML로 표현한 것이다. 한 개의 리스트와 세 개의 아이템으로 구성되어 있고, 아이템은 각각 Boolean, 4Byte unsigned Integer, ASCII의 자료형의 값을 취한다.

Ⅲ. 메시지 프로토콜 설계

아래 〈그림 4〉는 RUPI, SECS-II와 OPC 메시지의 구조를 도식화 한 것이다.

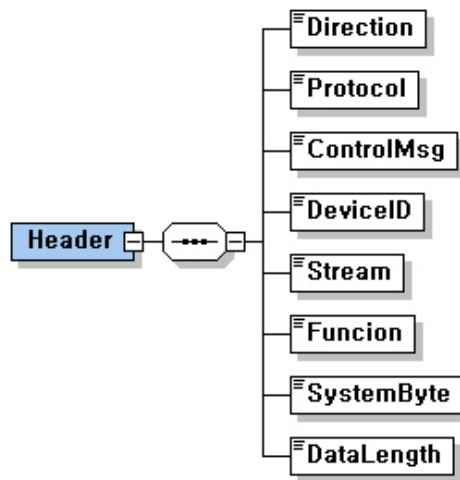


▶▶ 그림 4 RUPI, SECS-II, OPC 메시지 구조

위 그림에서 보는 바와 같이 RUPI의 메시지 구조는 OPC와 SECS-II보다 단순한 구조를 보이고 있다. 이것은 초기 PLC 메시지 형태와 유사한 것으로 URC 로봇이 증가하거나 기능이 확장되면 메시지 구조가 변경될 수 있음을 알 수 있다. 따라서 URC 로봇의 메시지는 기존에 앞서 개선된 장비 제어 메시지 구조와 URC 로

봇의 기능에 맞게 정의 되어야 할 필요가 있다. 각각의 메시지 구조는 트리형태로 매우 유사하며 RUPI는 트리 형태를 갖는 SECS-II와 다르고, SECS-II는 OPC와 그룹이 없이 List에 Item이 붙는 형식으로 이루어져 있다는 차이를 보이고 있다. 하지만 그 차이는 요소의 명칭에 관한 차이일 뿐 그 구조는 모두 트리 형태로 표현할 수 있다. 이것은 각 메시지를 하나의 메시지로 통합할 수 있다는 것으로 메시지가 하나로 통합되면 동일한 환경에서 모니터링 및 제어를 할 수 있다. 또한 IT 기술의 핵심인 웹서비스를 위하여 웹 문서의 형식을 표준화된 구조로 표현하는 XML(eXtensible Markup Language)을 사용하여 메시지를 구성하면 IT 기술과의 컨버전스가 용이하게 이루어 질 수 있다.

메시지를 통합하기 위하여 구조와 요소(element), 속성(attribute) 등의 관계를 정의하고 XML Schema를 설계하였으며, 작성된 XML Schema는 헤더와 데이터로 이루어져 있다. 〈그림 5〉는 메시지의 헤더를 표현한 것으로 〈표 3〉과 같은 요소를 갖는다.



▶▶ 그림 5. 헤더 부분 스키마

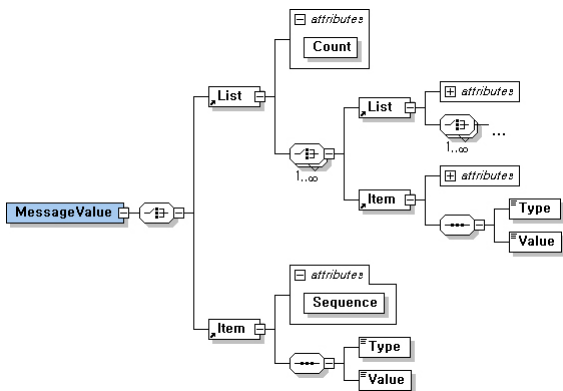
〈표 3〉에서와 같이 헤더의 요소는 각 메시지를 모두 포함하고 있음을 확인 할 수 있다.

데이터 부분은 실질적으로 장비에서 전송되는 데이터가 저장되는 부분이다. SECS-II 메시지는 다른 메시지에 비하여 복잡한 구조를 갖고 있다. 즉 SECS-II 메시지를 포함하는 구조를 만들면 RUPI, OPC 메시지를 포함할 수 있는 것이다. 또한 데이터 확장에 대한 것도 고려하여야 한다. 그러기 위해서는 SECS-II 메시지와 같은 트리 구조를 갖는 것이 적절하다.

표3. 각 요소와 의미

요소	RUPI	OPC	SECS-II
Direction	분산 로봇 플랫폼 미들웨어를 위한 요소	서버의 부하를 줄이는 분산 모니터링에 대비하여 역할별로 특화된 목적 서버 기록	
Protocol	IEEE802.11g	PLC	SECS-I, HSMS
ControlMsg	메시지임을 표시	ENQ, ACK, NAK	컨트롤 메시지일 경우 메시지 내용 기록
DeviceID	메시지를 전송한 장비의 ID		
Stream	요청식별자	주소	메시지의 Stream
Function	메시지타입 (call, reply, error, notify등)	Data type	메시지의 Function
SystemByte	메시지들의 구별을 위해서 필요한 바이트		
DataLength	메시지의 크기		

〈그림 6〉은 데이터 부분 스키마를 보여주고 있다. MessageValue 요소는 List 요소, Item 요소를 선택적으로 다수를 포함할 수 있는데, SECS-II는 여러개의 Item 또는 List가 존재하는 복합적 구조를 갖고 있기 때문이다. 이것을 명시하기 위해서 List는 자식 요소의 수를 저장하고 있는데 제안하는 구조에는 List 요소에서 Count 속성에 갖게 된다. 파싱하는 과정에서 List 요소에 명시된 Item의 개수를 확인하기 위하여 Item 요소는 Sequence라는 속성을 포함한다.



▶▶ 그림 6. 데이터 부분 스키마

표4. MessageValue 요소

요소	RUPI	OPC	SECS-II
List	확장영역	Group	List
Item	Item	Item	Item
Count	확장영역	Group의 개수	List의 개수
Sequence	확장영역	Item의 개수	Item의 개수

〈표 4〉의 MessageValue 요소에서 보듯이 RUPI는 Item 요소만을 사용하고 있으나 향후 데이터의 확장이 용이하다. OPC는 Group을 List으로 Item을 Item으로 적용되며 각 Count 및 Sequence는 Group과 Item의 개수를 의미한다. 이렇게 작성된 데이터 스키마는 각 메시지의 데이터 형식을 모두 만족 한다.

IV. 결론 및 향후과제

본 논문에서는 URC 로봇을 IT 기술과 컨버전스 하기 위한 메시지 프로토콜을 디자인 하였다. 기존의 장비 제어 메시지인 SECS-II와 OPC를 RUPI의 메시지와 비교 분석한 결과 RUPI 메시지는 기존 메시지의 초기 모습을 보여주고 있고 확장을 고려하지 않은 구조로 이 기종 로봇과의 메시지 교환에 있어서는 수정을 해야 하는 단점을 보여 주고 있다. 따라서, SECS-II와 OPC를 롤 모델로 메시지를 설계하였으며 XML을 기반으로 기존의 메시지도 포함하도록 하였다.

물론 SECS-II와 OPC 메시지가 IT 기술과 컨버전스 하기에 적절한 메시지는 아니지만 오랜 시간동안 각 분야에서 발전되어온 구조이므로 현재의 RUPI의 메시지 구조는 앞으로 유사한 형태로 발전될 가능성이 높다. 또한 기존 장비 메시지들은 IT기술과 컨버전스하기 위하여 메시지를 XML화 하기 위한 노력이 꾸준히 있어 왔다. 따라서 본 논문에서 제안한 메시지 프로토콜은 IT 기술과의 컨버전스를 원활하게 이룰 수 있는 기반기술로 활용 가능하다.

본 논문에서 제안한 메시지 프로토콜은 URC 로봇의 특징 및 기능을 모두 고려하지는 않은 상태로 지속적인 연구개발이 필요하며 제안한 메시지 프로토콜을 구현하

고 여러 상황을 적용 및 테스트하여 완성도를 높이는 것을 향후 목표로 한다.

■ 참고 문헌 ■

- [1] OPC Task Force "OPC(OLE for Process Control) Overview", October, 27, 1998.
- [2] Don W. Holley "Understanding and Using OPC for Maintenance and Reliability Applications", Computing & Control Engineering Journal Volume 15, Issue 1 pp 28-31, Feb-March 2004.
- [3] "OPC DA 3.00 Specification", March 2003, <http://www.opcfoundation.org>
- [4] SEMI Standards E5-0706, "SEMI Equipment Communications Standard 2 Message Content (SECS-II)", SEMI
- [5] 이승익, 장철수, 정승욱, 김중배, "로봇 소프트웨어 아키텍처의 연구 동향과 현황", 전자통신동향분석 제20권, 제2호, 2005.
- [6] 정승욱, 이승익, 김성훈, "네트워크 로봇을 위한 로봇 소프트웨어 플랫폼에 대한 연구", 정보과학회지 제26권, 제4호, 2008.
- [7] 임용묵, 황인수, 김우성, 박근덕 "반도체 장비의 메시지 통합을 위한 소프트웨어 구조 설계", 한국컴퓨터정보학회지 제12권, 제2호, 2007.
- [8] "지능형 로봇 플랫폼", 한국산업기술평가원, 산업기술동향분석, 2008.