

AI-Maker: 멀티 프로세스 컨트롤&모니터링 모델 지원 도구

심민석⁰, 박성규, 유대승, 김종환, 이명재

울산대학교 컴퓨터정보통신공학부

{sms, icoddy, yds, bearknight, ymj}@mail.ulsan.ac.kr

AI-Maker: A Tool for Supporting Multi Process Control Model

Minsuk Sim⁰ Sungkue Park Daesung Yoo Jonghwan Kim Myeongjae Yi
School of Computer Engineering & Information Technology, University of Ulsan

요 약

본 논문은 멀티 프로세스 컨트롤&모니터링 모델을 지원하는 AI-Maker 시스템에 대해서 기술한다. 멀티 프로세스 컨트롤&모니터링 모델은 제어 구조의 표준으로 떠오르고 있는 OPC와 마이크로소프트사의 분산형 보안 서비스 개념(하부 구조 서비스, 응용 프로그램 지원 서비스)들을 사용하여 실제, 가공, 그리고 가상 데이터 액세스 포인터들에 대해서 차별화 된 제어 구조를 제시한다. AI-Maker는 다양한 사용자에게 대한 차별화 된 서비스와 가공 또는 가상 액세스 포인터를 사용하기 때문에 장비 종속적인 시스템에 대한 개발비용을 감소시킬 수 있다. 또한 전문가 시스템의 학습에 필요한 표본 데이터를 제공하므로 필드버스 기반의 자동화 시스템 개발 및 유지보수 시 양질의 서비스를 제공할 수 있는 이점이 있다.

1. 서론

현대의 생산시스템은 제품에 대한 소비자들의 다양한 요구를 만족시키고, 생산기술의 변화와 컴퓨터 관련 기술의 급격한 발달에 빠르게 적응하기 위해서 유연성(flexibility), 통합성(integration) 및 동시성(concurrency)을 만족시키는 개방구조(open architecture)로의 전환이 요구되고 있다 [1][2]. 특히 제어 표준으로 떠오르는 OPC(OLE for Process Control)와 분산 시스템에 존재하는 다양한 리소스를 관리 및 제어하는 액티브 디렉토리[3] & Kerberos[4]와 같은 정보 통신 기술(Information & Communication Technology)이 밀접하게 결합되어 생산 시스템에 새로운 물줄기를 형성하고 있다.

OPC[5]는 마이크로소프트사의 WinSEM 그룹(Windows for Science, Engineering, and Manufacturing)으로 출발하여 OLE/COM 기술을 바탕으로 프로세스 데이터의 클라이언트 어플리케이션들과 서버(장비)들 사이의 인터페이스 방식을 규정한 것이다. [그림 1]은 OPC 서버의 작동 방식을 보여준다.

액티브 디렉토리(Active Directory Services)[3]는 네트워크에 산재해 있는 자원들을 찾아 이용할 수 있게 해주는 서비

스를 말하며 DNS(Domain Name System)와 X.500을 단순화한 LDAP(Lightweight Directory Access Protocol)[6] 기반으로 구성되어 있다. 더불어 비밀키 암호화에 기반을 둔 Kerberos와 연동되어 분산환경에서 강력한 서비스를 제공하고 있으며 본 논문에서는 분산된 사용자의 보안 정보에 관심을 가진다.

본 논문은 멀티 프로세스 컨트롤&모니터링 모델을 지원하는 AI-Maker(Advanced Instrument's Maker)시스템을 제공함으로써 쉽고 유연하게 분산 서비스들(액티브 디렉토리, 보안서비스)을 사용하고 OPC 기반의 장비 제어 시스템에 유연하고 확장성 있는 이점을 부여하고자 한다. AI-Maker가 제공하는 멀티 프로세스 컨트롤&모니터링 모델은 3-계층 구조(리소스, 로직, 그리고 프레젠테이션 계층)와 컴포넌트 개념을 기반으로 디자인하였기 때문에 엔터프라이즈 어플리케이션이 가지는 다양한 이점을 가질 수 있다. 리소스 계층은 다양한 OPC 기반 제어 서버 시스템에서 제공하는 실시간 데이터(액세스 포인터 데이터) 및 저장 데이터들과 쉽고 유연하게 결합할 수 있으며, 로직 계층은 리소스 계층에서 제공하는 리소스들(실시간 및 저장 데이터)에 대한 보안 설정을 제공하고 실제·가공·가상 데이터 제공 및 생성하는 역할을 제공한다. 프레젠테이션 계층은 XML 기술을 사용하여 콘텐츠와 뷰를 분리하고 접속 기기들의 타입에 맞는 뷰를 동적으로 결합하여 제공함으로써 다양한 타입의 모니터링 기기들과 유연하게 결합하고 통합할 수 있는 이점을 갖는다.

본 연구는 한국과학재단 지정 울산대학교 네트워크 기반 자동화 연구센터의 지원에 의해 이루어졌습니다.

논문의 구성을 살펴보면 2장에서는 관련 연구를 살펴보고 3장에서는 멀티프로세스 컨트롤&모니터링 모델과 AI-Maker의 처리 방법을 제시한다. 4,5장에서는 AI-Maker를 사용하여 제시 모델을 검증하고 6장에서는 결론 및 향후 연구 과제에 대해서 논한다.

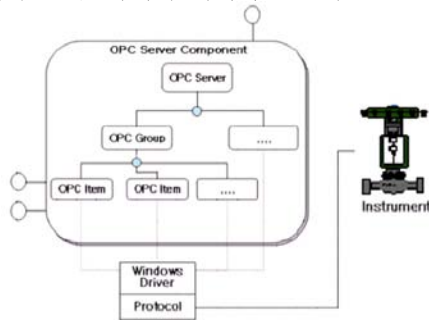


그림1 OPC 서버 구조

2. 관련연구

멀티 프로세스 컨트롤&모니터링 모델 또는 모델을 지원하는 시스템과 직접적인 연관이 있는 연구는 국-내외적으로 미비한 편이었다. 그리고 장비를 제어하는 연구 중 OPC와 같은 표준화 된 방법을 사용하는 연구들을 살펴보면 OPC를 직접적으로 이용하는 것과 컴포넌트의 래핑 기술을 사용하여 OPC를 래핑하고 특정 서비스를 추가하는 형태로 분류할 수 있었다.

OPC 컴포넌트를 직접적으로 이용하는 연구들은 대부분 시스템 통합 및 유지보수 비용을 낮출 수 있는 쪽에 초점을 두고 있다. 또한 XML 기술을 사용하여 유연성과 확장성을 장비 제어 시스템에게 부여하는 방향도 있다. Matthias[7], Wu[8]는 OPC에 대한 일반적인 개념을 소개하고 있으며, 특별히 Matthias[9]는 OPC기술을 사용하여 다양한 장비의 드라이버 통합구조를 제안하였다. Vassilis[10]는 OPC와 무선 인터넷을 연동하는 연구를 하였다.

OPC를 컴포넌트의 래핑 기술을 이용하는 연구들의 흐름은 대체적으로 XML 기술의 풍부한 표현 능력과 동반한다. 심[11]은 OPC-DA의 태깅 정보를 XML을 사용하여 래핑하고, 제어 로직을 생성하며, OPC 래퍼에서는 이를 분석하여 장비를 제어하는 연구를 하였다. 이외에 자바를 이용하여 사용자 인터페이스(UI)를 구성하고 JNI(Java Native Invocation) 기술을 사용하여 장비를 제어하는 방향의 연구 분야도 있었다.[12]

위와 같은 관련 연구 조사를 통하여 제어&모니터링에 관련 연구 동향이 XML 기반의 인터넷과 OPC 중심으로 흐르고 있고, 이러한 기술이 중요한 요소가 되고 있음을 알 수 있다. 또한 본 논문에서 제안하는 분산 개념을 적용한 장비제어 관한 적용 사례 및 연구는 빈약하였으며 특히 분산 개념의 응용 사례는 미비하였다.

3. 멀티 프로세스 컨트롤&모니터링 모델

본 논문에서 제안하는 멀티 프로세스 컨트롤&모니터링 모델은 [그림 2]에서 기술한 것과 같이 인터넷 프로토콜 기반 영역에 위치하며 [그림 3]과 같은 구조로 디자인하였다.

멀티 프로세스 컨트롤&모니터링 모델 아키텍처는 [그림 3]과 같으며 액세스 포인트를 분석하는 소스 분석 단계, 분석된 액세스 포인트를 탐색하고 가공하는 탐색·조작·생성 단계, 실제, 가공, 가상 액세스 포인트에 대한 보안 적용 단계, 네임스페이스를 이용하여 트리 구조화하는

과정, 그리고 액세스 포인트로부터 추출 또는 생성하는 데이터를 HMI 도구 또는 인터넷을 통하여 제공을 담당하는 프레젠테이션 단계로 구성한다.

모델 적용 환경

현재의 제어&모니터링 환경은 하드웨어 특성이 강한 필드버스 영역과 하위 영역의 데이터를 사용하여 의미 있는 정보를 추출하는 특성이 강한 어플리케이션 영역으로 분류한다. 필드버스 영역은 다양한 센서들을 통하여 상태 정보를 수신하고, 에뮬레이터 장비를 이용하여 특정 행위를 중점적으로 하며, 어플리케이션 영역은 인터넷을 기반으로 하위 레벨에서 발생하는 다양한 데이터를 수집 및 가공하여 의사 결정에 도움 주는 정보를 생성하는데 중점을 둔다.

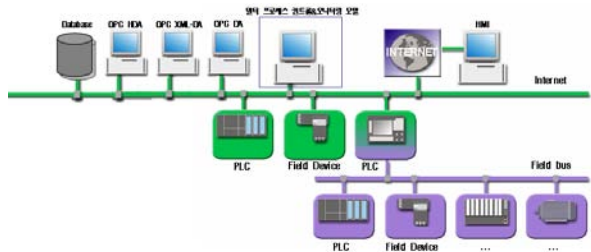


그림2 현재 제어&모니터링 시스템에서 제안 모델의 위치

필드버스의 구조는 크게 8개의 다양한 필드버스 프로토콜(Profibus, Foundation Fieldbus,...)로 분류하며 각기 마스터&슬레이브 구조로 이루어져 있다. 마스터(Master)는 대부분 PLC 장비 또는 마스터 카드를 가지는 컴퓨터가 되고 하위 레벨의 PLC와 연결되어 있는 형태를 가진다. 특히 마스터는 제어 로직을 가지며 로직에 맞게 행위를 한다. 슬레이브는 대부분 마스터 하위에 존재하는 PLC 장비들이며 다양한 아날로그 I/O 모듈, 디지털 I/O 모듈을 플러그인 할 수 있다. 모듈의 I/O 단자에 다양한 종류의 디바이스(센서와 에뮬레이터)가 결합되어 있고 마스터에 있는 제어 로직을 통하여 동작한다. 특히 [그림 2]에서 인터넷 라인의 통신을 위하여 프로피넷(PROFINET)모듈을 통하여 필드 버스 레벨의 제어 및 모니터링 정보를 인터넷 기반의 공장정보관리(Plant Information Management), 데이터 분석과 통합기능을 제공하는 운영 정보 시스템(Operation Information System), 생산 시스템(Manufacturing Execution system)...등에 대한 연결 역할을 한다. 그리고 디바이스의 상태 및 컨트롤 정보를 담당하는 OPC DA(Data Access), DA 컴포넌트들의 수평적인 통합을 제공하는 OPC DX, 이전에 저장된 정보의 관리를 담당하는 OPC HDA, 그리고 웹서비스 기반으로 상태 및 컨트롤 정보를 담당하는 OPC XML-DA 등의 컴포넌트들이 상위 영역에 존재하는 구조로 되어 있다.

소스 분석 단계

소스 분석단계는 모델에서 사용하는 입력 소스의 제어 포인트를 분석하여 모델 내부에서 사용 가능한 형태로 변환하기 위한 단계이다. 입력 소스의 제어 포인트의 영역은 디바이스 액세스 포인트와 데이터 베이스(본 논문에서는 SQL Server 2000 기준)에 저장된 정보를 가리키는 액세스 포인트로 분류한다.

디바이스 액세스 포인트는 장비에 부착되어 있는 센서 또는 에뮬레이터의 접점으로부터의 데이터 소스이고 OPC

DA 3.0 방식의 인터페이스를 이용하여 OPC DA 서버 내부에 구성된 액세스 포인트 정보를 추출하도록 디자인하였다. [그림 1]에서 보이는 OPC DA서버를 구성하는 태그 정보(OPC Item)가 현재 디바이스의 상태 정보 포인트와 1:1 대응한다.

데이터 베이스 액세스 포인트는 데이터 베이스의 특정 테이블의 필드와 매칭되며, 데이터 소스 추출하는 방법은

하는 데이터와 장비의 즉각적인 값을 요구하고 수집하는 것을 기준으로 정의하였다. 실제 액세스 포인트는 OPC 인터페이스를 사용하여 장비와 연결되어 있는 OPC 서버로부터 2가지 형태정보를 요구하여 처리하였다. 가상 액세스 포인트는 타이머를 두고 일정 시간 간격으로 테이블의 필드의 내용을 장비의 데이터처럼 보내고 항상 보낸 정보의 데이터를 캐쉬에 저장하여 즉각적인 상태 값의 경우에

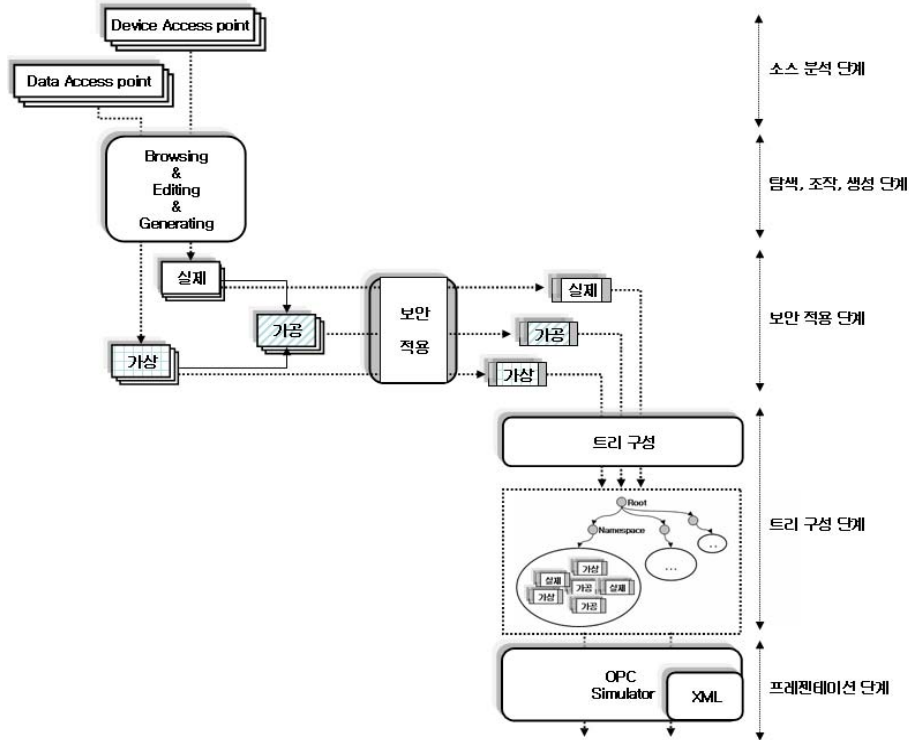


그림 3 멀티 프로세스 컨트롤&모니터링 모델

OPC Historial를 사용하는 방법과 마이크로소프트사의 SQL 서버를 사용하는 방법으로 나눌 수 있다. OPC Historial 데이터는 OPC Foundation에서 제공하는 표준 인터페이스를 사용하였고, SQL 서버를 사용하는 방법은 SQL 서버 내부의 특정 데이터베이스의 사용자 정의 테이블의 내용과 테이블을 구성하는 필드의 정보를 추출하는 질의를 사용하였다. 아래의 질의(SQL) 중 첫 번째 질의는 사용자 정의 테이블 리스트를 반환하는 질의이며 나머지 하나는 테이블 정보를 받아서 테이블을 구성하는 필드의 리스트 및 속성을 가져오는 프로시저문이다.

```
select o.name from sysobjects o
where o.xtype = 'u' and permissions (o.id)&4096 <> 0

Exec sp_columns @table_name
```

탐색 · 조작 · 생성 단계

탐색 · 조작 · 생성 단계는 소스 분석단계에서 추출한 데이터 액세스 포인트를 이용하여 새로운 데이터 액세스 포인트를 생성 및 가공하는 과정이다. 특별히 필드버스에 접점을 가지고 있는 액세스 포인트를 "실제", 데이터 베이스의 테이블을 구성하는 필드로부터 데이터를 생성하는 액세스 포인트를 "가상", 그리고 실제 액세스 포인트와 가공 액세스 포인트로부터 추출하는 데이터를 조작하여 가공 정보를 생성하는 액세스 포인트를 "가공"이라는 용어로 정의하고 사용하였다.

데이터 액세스 포인트로부터 데이터 수신의 기본 룰은 필드 버스의 기본 방식과 같이 일정 시간 간격으로 발생

캐쉬에 저장되어 있는 데이터를 반환하는 방법을 사용하였다. 가공 액세스 포인트는 실제 액세스 포인트의 데이터 값과 가상 액세스의 데이터 값을 이용하여 처리하였다.

보안 적용 단계

보안 적용 단계는 도메인 컨트롤러의 분산형 보안 서비스 구조를 추출하여 액세스 포인트들에 대한 보안 모델을 적용하는 하는 과정이다. 아래의 [그림 4]는 여러 제어 컴퓨터를 하나의 도메인 컨트롤러를 묶고 여러 컴퓨터의 자원 관리를 위하여 액티브 디렉토리의 조직구성단위(OU) 개념을 사용하여 분류한 모습이다. 이런 각각의 분류에 사용자 또는 그룹의 사용권한을 정의한다. 그러면 도메인 컨트롤러 제어 하에 있는 개별 컴퓨터에 로그인한 사용자는 로그인 정보에 의하여 여러 액세스 포인트의 권한을 가지는 스마트한 구조를 제공한다.

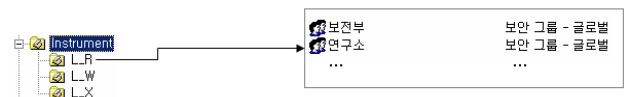


그림 4 보안 모델 구조의 예

트리 구성 단계

트리 구성단계는 위의 단계에서 구성한 3가지 타입의 액세스 포인트의 구조를 조직화하는 단계이다. 특별히 최상위의 루트를 기준으로 다양한 네임스페이스를 만들고

네임스페이스 내부에는 3가지타입의 많은 액세스 포인터를 포함할 수 있다.

프레젠테이션 단계

프레젠테이션 단계는 모델의 내부와 외부를 연결하는 방법을 정의한 곳이다. 특히 OPC 통신 방법을 사용하고 있는 기존의 HMI 도구들을 위하여 기본 보안을 적용하여 제공하는 OPC Simulator와 보안을 제공하기 위하여 클라이언트 측에 OPC 시뮬레이터를 만들어주고 모델 내부와 RPC 통신을 하여 처리하는 방법을 제공한다. 또한 도메인 외부의 접속 사용자를 처리하기 위하여 웹 서비스와 XML 기술을 사용한다.

4. AI-Maker 시스템 사용자 인터페이스

아래의 [그림 5]는 SQL서버에 접속하여 RData 필드를 가상 액세스 포인터로 변환하는 모습이다.

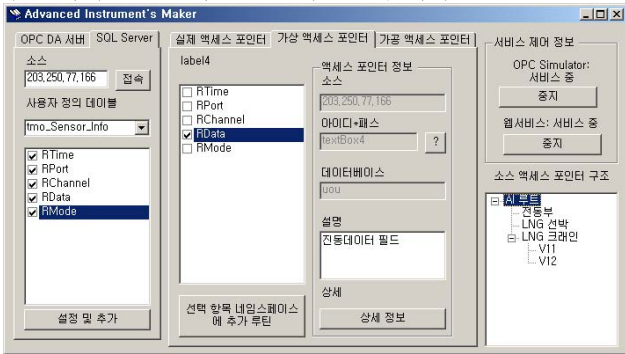


그림 5 AI-Maker 주 실행화면

5. 테스트

멀티프로세스 컨트롤 & 모니터링 모델과 AI-Maker 시스템의 효율성을 테스트하기 위하여 Siemens에서 제안하는 ProfiBus 방식의 필드버스 환경을 이용하였다. 세부적으로 Siemens PLC S300 2대를 마스터&슬레이브로 연동하였고 마스터에는 프로피넷 모듈을 통하여 이더넷과 통신한다. 슬레이브에 디지털 I/O 모듈을 추가하고 센서를 설치하였다. 그리고 제안 모델은 OPC DA 서버 위치에 본 논문에서 제안하는 모델을 설치하였다.

테스트는 같은 도메인 내부의 액티브 디렉토리에 등록되어 있는 리모트 컴퓨터의 사용자가 테스트하는 것과 WAP 애플레이터를 사용하여 검증하는 방법을 사용하였다. 아래의 [그림 6]는 WAP 애플레이터를 통한 결과이다.

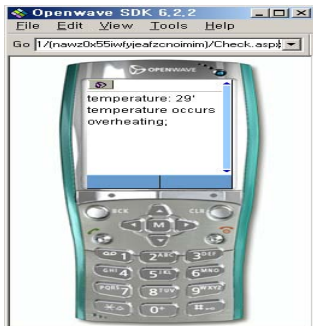


그림 6 WAP 애플레이터 실행

6. 결론 및 향후 연구 과제

멀티 프로세스 컨트롤&모니터링 모델은 3-계층 기반의 구조를 기본으로 하며 실제 데이터(필드 디바이스의 액세스

스 포인터에 연결된 데이터)와 가상 데이터(데이터베이스 및 OPC HDA)를 이용하여 새로운 가공 데이터(데이터 액세스 포인터)를 생성한다. 루트와 네임스페이스 노드를 기준으로 3가지 타입의 데이터 포인터를 재구성하고, 액티브 디렉토리 서비스와 Kerberos가 가지고 있는 분산된 자원(특히 사용자)에 대한 보안 정보를 사용하여 액세스 포인터에 대한 차별적인 권한을 부여하였다. 이러한 특징으로 인하여 원격에서 접속하는 사용자에게 대해서 차별적인 컨트롤 및 모니터링 기능을 제공함으로써 성능을 향상시킬 수 있고 유지 보수에 대한 비용을 줄일 수 있다. 또한 웹기반 제어 및 모니터링으로의 변화에 있어서 중요한 기술이 될 것이라 생각된다.

향후 연구 과제로는 다양한 장비에 대한 검증 작업과 확대 적용에 관한 연구가 수행되어야 한다.

참고문헌

- [1] W.Sperling and P.Lulz, "Enabling open control systems: An introduction to the OSACA system platform", ESPRIT III Project: Stuttgart: FISW GmbH, 1995
- [2] P.K. Wright, "Principles of open-architecture manufacturing", Journal of Manufacturing System, vol. 14, no.3, pp. 187-202, 1995
- [3] Microsoft, <http://www.microsoft.com/windowsserver2003/technologies/directory/activedirectory/default.mspx>
- [4] Microsoft, <http://www.microsoft.com/windowsserver2003/technologies/security/default.mspx>
- [5] OPC Foundation, <http://www.opcfoundation.org>
- [6] Microsoft, <http://www.microsoft.com/technet/prodtech/nolwinntas/maintain/adsildap.mspx#XSLTsection128121120120>
- [7] Matthias Riedl, Mario Thron, Thomas Hadlich, "DriveServer-significantly reduce in engineering expense", The 27th Annual Conference of the IEEE Industrial Electronics Society
- [8] Wu Sitao, Qian Qingquan, "Combing OPC with Autonomous Decentralized Systems", 2000 IEEE
- [9] Matthias Riedl, Mario Thron, Thomas Hadlich, "DriveServer-significantly reduce in engineering expense", IECON'01: The 27th Annual Conference of the IEEE industrial Electronics Society
- [10] Vassilis, OPC-SMS: a wireless gateway to OPC-based data sources, Computer Standards & Interfaces 24 (2002) 437-451
- [11] Minsuck Shim, Sungkyu Park, Dae-Sung Yoo, Jong-Hwan Kim, Myeongjae Yi, "A Study on the Flexible and Efficient Instrument Control Software Generation", Proceedings of the 7th Korea-Russia International Symposium, KORUS 2003, vol2. pp.447-452
- [12] Harald Kleines, "Access to Industrial Process Periphery Via java for Process Control(JFPC)", IEEE TRANSACTIONS ON NUCLEAR SCIENCE, VOL.49, NO.2 APRIL 2002
- [13] Wireless Application Protocol Architecture Specification, WAP-210-WAPArch-20010712, Version 12-July 2001.