

Java2OPC Gateway: 다양한 플랫폼에서의 OPC 접근

유대승, 심민석, 박성규, 박재희, 김종환, 이명재
울산대학교 컴퓨터정보통신공학과
e-mail:bearknight@nate.com

Java2OPC Gateway: OPC Access in Various Platform

Dae-Sung Yoo*, Min-Suck Sim*, Sung-Ghwe Park*, Jae-Hee Park*, Jong-Hwan Kim*, Myoung-Jae Yi*

*School of Computer Engineering & Information Technology,
University of Ulsan

요 약

본 연구에서는 OPC(OLE for Process Control)가 마이크로소프트의 COM(Component Object Model)과 DCOM(Distributed COM) 기술을 사용함으로써 윈도우즈 플랫폼에서 클라이언트 어플리케이션이 개발되어 제어 하는 제약을 해결하기 위하여 Java2OPC Gateway를 제안한다. Java2OPC Gateway는 프로세스 컨트롤 장비와 연결된 OPC 서버쪽과 통신하는 클라이언트의 기능과 다양한 플랫폼에서 프로세스 데이터에 접근하기 위한 자바 어플리케이션과 통신하는 서버의 기능을 가진다. 클라이언트 모듈은 C로 작성되고 서버 모듈은 Java로 작성되어 클라이언트 모듈과 서버 모듈은 JNI(Java Native Interface)로 연결 된다.

1. 서론

복잡한 설비들을 제어하기 위하여 장비들을 직접 제어하기보다는 PC 기반으로 제어하게 됨에 따라 이런 자동화 장비들의 제어 및 모니터링을 위한 소프트웨어의 중요성이 점차 증대되고 있다.

그리고 PC 기반의 제어 시스템은 중앙 집중적인 제어 방식에서 분산 제어 시스템으로 변화되어 가고 있다. 이로 인하여 제어 소프트웨어는 개방성, 상호 운용성, 네트워크 지원 등이 요구된다.

그러나 PC 기반의 제어 소프트웨어들은 많은 다른 벤더들에 의해서 호환성을 고려하지 않고 개발되어 지므로 상호 운영성이 부족하여 다른 벤더들에 의해서 개발된 제어 소프트웨어들을 통합하기는 매우 어렵다. 따라서 이러한 제어 소프트웨어 컴포넌트들을 통합하기 위한 표준 소프트웨어 인터페이스는 매우 중요한 요소가 된다.

프로세서 제어에 대하여 개방성, 상호 운용성, 호

울성, 유지보수성 등을 제공하는 표준 인터페이스를 정의하는 OPC는 산업현장의 제어 분야에서 많은 연구가 진행되고 있으며 사용이 점차 확산되고 있다.

OPC(OLE for Process Control)는 프로세스 제어 분야에서 마이크로소프트의 COM(Component Object Model)과 DCOM(Distributed COM) 기술을 사용하여 각종 어플리케이션들이 여러 종류의 프로세스 컨트롤 장비(DCS, SCADA, PLC 등)들로부터 데이터를 수집하고 조작하는 개방형 표준 인터페이스를 정의한다.

OPC 표준은 프로세스 데이터의 클라이언트 어플리케이션들과 서버들 사이의 인터페이스 방식을 규정한 것으로 서로 다른 벤더들에 의해 공급되는 클라이언트 어플리케이션과 서버 어플리케이션들 사이에 높은 수준의 상호 호환성 있는 정보 처리를 제공한다.

그러나 OPC는 마이크로소프트의 COM, DCOM 기술을 사용함으로써 윈도우즈 플랫폼에서 클라이언트 어플리케이션이 개발되어 저야 하는 제약을 가진다.

본 연구에서 제안하는 Java2OPC Gateway는 OPC 서버와 OPC 서버에 접근하는 클라이언트 어플리케이션 사이의 미들웨어이다. COM, DCOM을 지원하지 않는 플랫폼에서 개발된 클라이언트 소프트웨어는 OPC 서버에 직접적으로 연결할 수 없지만 제안하는 미들웨어를 통해서 OPC 서버에 접근할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 OPC와 관련된 기존의 연구들에 대하여 간략하게 소개하고, 3장에서는 Java2OPC Gateway의 기술적 배경에 대해서 기술한다. 4장에서는 Java2OPC Gateway의 시스템구성, 기능적 요구사항, 프로세스 데이터 접근 과정에 대해 설명한다. 마지막으로 5장에서는 결론 및 향후 연구 과제를 기술한다.

2. 관련연구

자동화 분야에서 OPC는 필드버스와 함께 최근에 가장 많은 관심을 가지게 되는 분야 중 하나로 처음 그 명세가 발표된 이후로 지금까지 다양한 연구들이 진행되고 있으며, HMI(Human Machine Interface), SCADA(Supervisory Control and Data Acquisition), DCS(Distributed Control System) 등과 같은 PC 기반의 자동화 시스템 분야의 장비 및 소프트웨어 벤더들이 대부분 그들의 제품에 OPC 클라이언트와 OPC 서버 인터페이스를 제공하고 있다[1].

본 논문에서는 OPC에 대한 기존의 연구를 다음과 같은 세 가지 분야로 나누고, 각 분야에 해당하는 연구들을 간략하게 소개하도록 하겠다.

a. 기존의 레거시 시스템에 OPC 표준을 적용

OPC 표준을 적용하는 연구들 중에는 기존의 DCS, PLC, SCADA 등의 레거시 시스템에 OPC 기술을 적용해서 “Plug and Play” 형식으로 통합함으로써 개방성, 상호 운용성, 유지보수성 등을 향상시키는 방법[2][3]과 OPC 표준을 기반으로 여러 필드버스 시스템 객체들을 공통 객체들로 정의함으로써 다양한 필드버스 시스템들(LON, EIB, EHS, PROFIBUS 등)을 통합하는 방법에 대한 연구[4] 등이 있다.

b. OPC 표준 명세를 활용

드라이버 벤더들로 구성된 DRIVECOM 사용자

그룹에 의해서 OPC 표준을 기반으로 하고 XML 기반의 디바이스 기술문서를 사용하여 드라이버들간의 표준적인 통신 인터페이스에 대한 표준 명세라고 할 수 있는 DriveServer 개발에 대한 연구[5]와 분산된 자동화 시스템에서 서로 다른 어플리케이션간의 효율적인 정보 교환을 위해서 OPC 표준 명세를 포함하는 Data Engine(DE)이라는 개념을 소개하는 연구[6] 등이 있다.

c. OPC에 대한 기타 응용

OPC에 대한 개념을 활용 또는 확장하는 것으로 OPC 아이디어를 기반으로 JAVA의 JNI를 이용하여 직접 실시간으로 프로세스를 제어하는 방법에 대한 연구[7]와 Web Gateway를 통해서 OPC를 서비스하는 방법에 대한 연구[8] 등이 있다.

OPC 관련 제품들은 OPC Foundation[9]에서 찾아볼 수 있으며 Client, Server, Developer Tools & Services로 분류되어 많은 제품들이 소개되어 있다.

기존의 연구들과 제품들은 COM, DCOM을 사용할 수 있는 윈도우즈 플랫폼을 기반으로 하는 연구들인 반면 본 연구에서 제안하는 Java2OPC Gateway는 윈도우즈가 아닌 플랫폼에서 OPC를 접근하는 방법에 대한 연구라고 할 수 있다.

3. 기술적 배경

3.1 OPC(OLE for Process Control)

OPC 1996년 발표된 이후 프로세스 데이터의 클라이언트 어플리케이션(OPC Client)들과 서버(OPC Server)들 사이의 인터페이스 방식을 규정하는 표준으로 자리 잡고 있다. 다음의 그림 [1]은 OPC 표준의 클라이언트와 서버간의 시스템 구조를 도식화한 것이다.

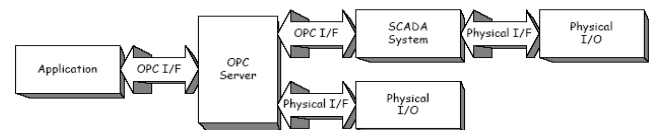


그림 1 OPC Client/Server Relationship

OPC는 서로 다른 여러 종류의 시스템들에 대한 상세한 기술적인 사양에 대하여 클라이언트 소프트웨어를 독립시켜 준다. 이로 인하여 클라이언트 어플리케이션 개발자들은 프로세스 컨트롤 장비로부터 데이터를 접근하기 위하여 단지 하나의 인터페이스 드라이버만 개발하면 된다. 또한 프로세스 컨트롤 장비를 개발하는 벤더는 클라이언트 소프트웨어에 영향을 주지 않고 OPC서버 인터페이스를 유지하면

서 여러 가지 기능들에 대한 수정 및 변경을 할 수 있다는 장점을 가진다. OPC 사용자가 가질 수 있는 이점으로는 사용자의 필요에 가장 적합한 클라이언트 어플리케이션을 선택할 수 있다는 것이다.

3.2 JNI(Java Native Interface)

JNI는 Java와 다른 언어로 작성된 어플리케이션과 상호작용을 가능하게 하기 위해 제공되는 인터페이스이다.

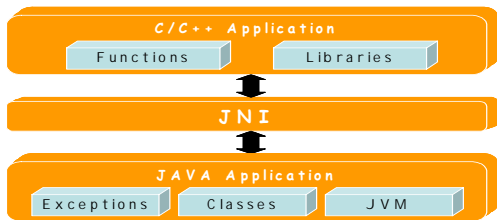


그림 2. JNI(Java Native Interface)

그림 [2]와 같이 JNI는 C/C++ 어플리케이션과 Java 어플리케이션 사이의 다리 역할을 한다. JNI를 사용하게 되면 다른 언어의 장점을 이용할 수 있으므로 Java에서 구현할 수 없는 기능을 구현할 수 있다는 장점을 가진다. 반면 JNI를 사용함으로써 Java가 가지는 플랫폼 독립성을 잃어버린다는 것과 언어별 데이터형의 변환과 예외 처리에 많은 신경을 써야 하는 단점도 존재한다. 그러나 Java의 장점을 가지고 하드웨어에 접근하기 위한 방법으로 JNI는 매우 중요한 위치를 차지한다.

4. Java2OPC Gateway

4.1 Java2OPC Gateway 시스템 구성

본 논문에서 제안하는 Java2OPC Gateway의 시스템 구성은 그림 [3]과 같다.

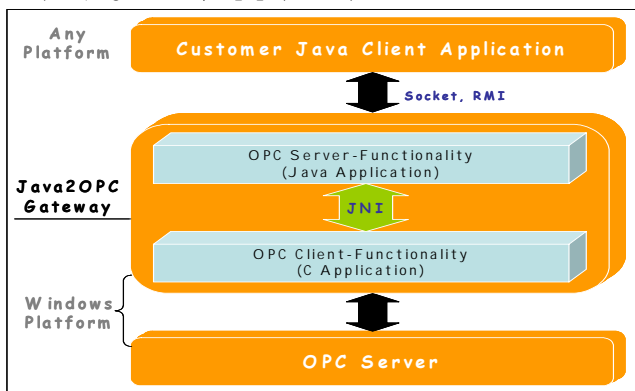


그림 3. Java2OPC Gateway Architecture

Java2OPC Gateway는 OPC 서버로의 접근을 위해 윈도우즈 플랫폼에서 실행되어야 하며, OPC 서버와 같은 노드에 있거나 다른 노드에 있을 수 있

다.

Java2OPC Gateway는 OPC 서버와의 관계에서는 OPC 클라이언트의 역할을 수행하고 클라이언트 어플리케이션과의 관계에서는 OPC 서버의 역할을 수행한다. 따라서 Java2OPC Gateway는 OPC 서버로서의 기능과 OPC 클라이언트로서의 기능을 모두 가져야한다.

OPC 서버의 기능을 수행하는 모듈은 Java로 구현되므로 클라이언트 어플리케이션을 Java로 구현하는 경우 RMI(Remote Method Invocation)와 같은 분산 기술을 사용할 수 있다. 또한 OPC 명세의 Complex Data Type에 대한 구현이 쉽고 Java의 "Write Once Run Anywhere"라는 장점을 가질 수 있다.

OPC 클라이언트의 기능을 수행하는 모듈은 COM, DCOM과 JNI를 지원하는 C 언어로 사용해서 OPC 표준 명세를 따르도록 구현한다.

두 모듈은 JNI를 통해서 상호작용하며 OPC 서버와 클라이언트 어플리케이션의 중간에 위치해서 COM, DCOM을 지원하는 윈도우즈가 아닌 플랫폼에서 개발된 클라이언트 어플리케이션이 OPC 서버로의 접근을 가능하게 한다.

4.2 Java2OPC Gateway의 기능적 요구사항

Java2OPC Gateway의 기능적 요구사항은 OPC 클라이언트로서의 요구사항과 OPC 서버로서의 요구사항으로 나눌 수 있으며, 현재로는 OPC DA(Data Access)에 대한 요구사항만을 고려한 것이다.

OPC 서버로서의 기능적 요구사항은 클라이언트 어플리케이션에 대하여 OPC 서버 기능을 수행하는 것으로 OPC 그룹 객체의 생성과 조작, 그룹 내에서 접근하고자 하는 data(item)를 구성, 그룹을 activated 또는 deactivate 상태로 설정, 동기적 접근과 비동기적 접근을 제공하는 등의 기능을 가져야 한다.

OPC 클라이언트로서의 기능적 요구사항은 OPC 서버에 대하여 클라이언트의 기능을 수행하는 것으로 모든 클라이언트 어플리케이션의 요청을 실제 OPC 서버에 요청하고 그 결과를 되돌려 줄 수 있어야 하다. 또한 비동기적 접근을 지원하기 위해서는 OPC 서버에게 콜백 인터페이스를 노출시켜야 한다.

4.3 Java2OPC Gateway를 이용한 DA(Data Access)

클라이언트 어플리케이션이 프로세스 데이터에 접근은 동기적 접근과 비동기적 접근으로 나눌 수 있으며 읽기(read), 쓰기(write), 전체 새롭게 읽기

(refresh), 구독하기(subscribe) 등이 있다.

그림 [4]는 동기적 읽기에 대한 과정을 시퀀스 다이어그램으로 보인 것이다.

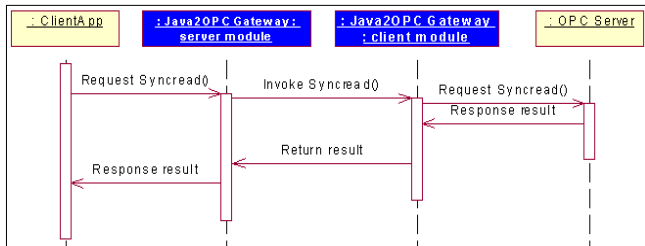


그림 4. Synchronous Read

동기적 읽기는 비동기적 읽기에 비해서 비교적 간단하게 수행된다. 클라이언트 어플리케이션이 동기적 읽기 요청을 요구하면 Java2OPC Gateway의 서버 모듈이 이 요청에 해당하는 클라이언트 모듈의 메시지를 JNI 방식으로 호출하게 되고 클라이언트 모듈이 OPC 서버에 최종적으로 동기적 읽기를 요청하게 된다. 요청에 대한 결과는 역방향으로 클라이언트 어플리케이션에게 전달되어 진다.

그림 [5]는 비동기적 읽기에 대한 과정을 시퀀스 다이어그램으로 보인 것이다.

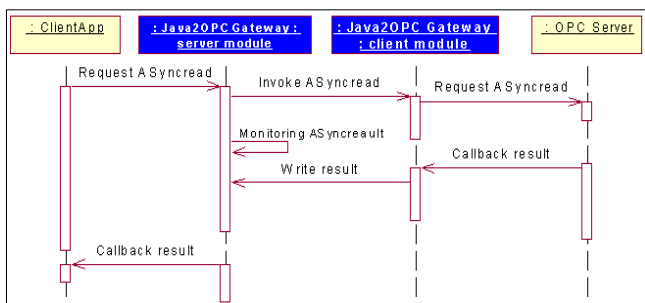


그림 5 Asynchronous Read

비동기적 읽기는 콜백 인터페이스의 호출과 호출에 의해서 수행된다. 그러나 Java2OPC Gateway의 서버 모듈과 클라이언트 모듈은 JNI 방식을 사용하므로 콜백 인터페이스의 호출과 호출을 사용할 수 없다. 따라서 비동기적 읽기 요청에 대하여 서버 모듈은 이 요청에 해당하는 클라이언트 모듈의 메시지를 호출 후 요청에 대한 결과 값이 비동기 요청을 위한 저장소에 기록되는 것을 지속적으로 감시한다. 요청한 결과 값이 기록되어 지면 최종적으로 클라이언트 어플리케이션의 콜백 인터페이스를 호출하여 결과 값을 클라이언트에 전달하게 된다.

5. 결론

본 연구에서 제안한 Java2OPC Gateway는 OPC

서버와 클라이언트 어플리케이션 사이의 미들웨어로 COM, DCOM을 지원하지 않는 플랫폼에서 개발된 클라이언트 소프트웨어가 이 미들웨어를 통해서 OPC 서버에 접근할 수 있도록 한다.

Java2OPC Gateway는 OPC 서버로서의 기능과 OPC 클라이언트로서의 기능을 모두 가져야한다. 클라이언트 기능을 수행하는 모듈은 C로 작성되고 서버 기능을 수행하는 모듈은 Java로 작성되어 클라이언트 모듈과 서버 모듈은 JNI(Java Native Interface)로 연결된다.

현재 Java2OPC Gateway는 OPC 명세들 중 DA(Data Access)만을 고려해서 설계되었고, 향후 다양한 OPC 표준 명세들(Alarms & Events, Historical Data Access, Security, Batch, Data Exchange 등)을 지원할 수 있도록 확장하도록 하겠다.

참고문헌

- [1] Li Zheng, Hiroyuki Nakagawa "OPC(OLE for Process Control) Specification and its Eevelopments" SICE 2002. Proceedings of the 41st SICE Annual Conference , Volume: 2 , 5-7 Aug. 2002 Pages:917 - 920 vol.2
- [2] MARK JANKE "OPC-Plug and Play Integration to Legacy Systems" Pulp and Paper Industry Technical Conference, 2000. Conference Record of 2000 Annual , 19-23 June 2000 Pages:68 - 72
- [3] Wu Sitao, Qian Qingquan "Using Device Driver Software in SCADA Systems" Power Engineering Society Winter Meeting, 2000. IEEE , Volume: 3 , 23-27 Jan. 2000 Pages:2046 - 2049 vol.3
- [4] Tomas Bangemann, Jörg Hähnliche Peter Neumann "Integration of Fieldbus Systems in Computer-Aided Facility Management" Industrial Electronics Society, 1998. IECON '98. Proceedings of the 24th Annual Conference of the IEEE , Volume: 3 , 31 Aug.-4 Sept. 1998 Pages:1835 - 1840 vol.3
- [5] Matthias Riedl, Mario Thron, Thomas Hadlich "DriveServer-significantly reduce in engineering expense" Industrial Electronics Society, 2001. IECON '01. The 27th Annual Conference of the IEEE , Volume: 1 , 29 Nov.-2 Dec. 2001 Pages:285 - 288 vol.1
- [6] Xu Hong, Wang Jianhua, Zhang GuiQing, Zheng ShiQuan "Using Data Engine in Distributed Automation System" Power System Technology, 2002. Proceedings. PowerCon 2002. International Conference on , Volume: 3 , 13-17 Oct. 2002
- [7] Harald Kleines, Peter Wüstner, Kirsten Settke, Klaus Zwoll " Access to Industrial Process Periphery Via Java for Process Control (JFPC)" Nuclear Science, IEEE Transactions on , Volume: 49 , Issue: 2 , April 2002 Pages:465 - 468
- [8] Vassilis KAPSALIS, Konstantinos CHARATSIS, Athanasios P.KALOGERAS, George, PAPADOPOULOS "Web Gateway: A Platform for Industry Service over Internet" Industrial Electronics, 2002. ISIE 2002. Proceedings of the 2002 IEEE International Symposium on , Volume: 1 , 8-11 July 2002
- [9] OPC Foundation, "http://www.opcfoundation.org"