시뮬레이션 모델과 외부 Client 간의 연동을 위한 OPC Server 개발

박신열*, 이명수, 서인용, 홍진혁
한전 전력연구원

The Development of OPC Server for Communication between Simulation Model and External Application

Shin-Yeol Park, Myeong-Soo Lee, In-Yong Seo, Jin-Hyuk Hong

Abstract

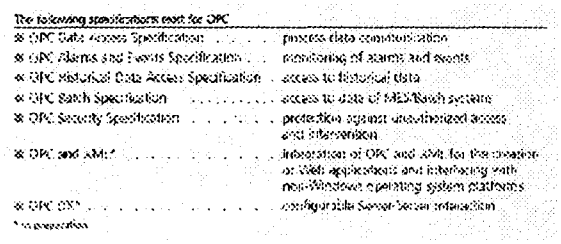
전력연구원에서는 울진 표준형원전의 시뮬레이터를 개발하였으며, 여기에는 가상 주제어실, 발전소 현상 감시, 중대사고 등 많은 클라이언트 프로그램들이 있다. 이러한 프로그램들은 시뮬레이터와 연동되어 필요한 값을 읽고 쓰는 과정이 필요하다. 그러나 현재는 해당 시뮬레이터 환경에서 개발되지 않는 외부의 응용프로그램이 이들 시뮬레이터의 값들을 엑세스하는 방법은 제공하지 않고 있다. 본 보고서에서는 이러한 문제점을 해결하기 위해 외부 프로그램들이 시뮬레이션모델의 각종 변수들을 효과적으로 엑세스하여 값을 Read/Write 할 수 있는 OPC (OLE for Process Control) Server를 개발하였다. 본 프로그램은 Rockwell의 OPC Server Toolkit인 RSI OPC/DDE Server Toolkit Library를 이용하여 개발하였으며 시뮬레이터가 실행되고 있는 컴퓨터에서 실행된다. 본 보고서에서는 OPC 일반개념, 개발한 OPC Server의 소개, OPC Server의 적용결과 등을 기술하고자 한다.

1. 서론

OPC는 OLE for Process Control의 약자로서 automation/control 응용프로그램, field systems/device 및 business/office 응용프로그램간의 호환을 위한 마이크로소프트의 OLE, COM, DCOM 기술을 기반으로 한 표준화된 API 집합이다. OPC는 하나의 프로그램이 다른 프로그램의 인터페이스를 이용하여 그로부터 정보를 얻게 해주는 소프트웨어 아키텍처인 COM에 기반하고 있다. OPC는 서버/클라이언트간 원리를 준수하며, 동시에 다양한 서버와 클라이언트간의 통신을 가능하게 한다. 현재는 DCOM에 기반을 두고 있지만 향후에는 Web Services를 기반으로 할 것이다. 즉, OPC는 Devices, Databases and Applications 간의 통신을 해결하는데 적당한 인터페이스를 제공한다.

1.1 OPC Specification

OPC Data Access Server에서는 OPC는 클라이언트를 위해 process 데이터를 유지하며, 클라이언트는 데이터를 read/write 혹은 subscribe 할 수 있다. 아래의 표는 OPC Data Access 스펙 이외에도 여러 스펙을 보여 주고 있다



1.2 Types of Data Exchange

OPC는 Data Access를 위해 Synchronous call, Asynchronous call, Refresh, Subscription 등 4종류의 통신 메커니즘이 고안되었다. 클라이언트가 동기 읽기 신호를 발하면 서버는 요청한 값이 반환되기 전까지 호출 thread에 제어권한을 주지 않는다. 이때 동기 쓰기는 가능하다. 클라이언트가 비동기 읽기 신호를 발하면 서버는 호출 thread에 즉시 컨트롤을 반환하고 후에 사전 정의된 통신 경로를 통해 요청한 값을 전송한다. 이때 비동기 쓰기는 가능하다. 동기, 비동기 호출을 하려면 읽거나 써야 할 값의 목록을 클라이언트가 지정해야 한다. 이 메커니즘은 polling에 의해 작동하고 클라이언트가 가끔 기기에 액세스해야 할 때 사용된다. Refresh와 Subscription은 callback 메커니즘으로써 사전 정의된 데이터 포인트 집합에 액세스하기 위해 사용 된다. Refresh는 "pull" 메커니즘이고, Subscription은 "push" 메커니즘이다. 클라이언트가 Refresh 호출을 발하면 OPC Server는 사전 정의된 통신 경로를 이용하여 사전 정의된 데이터 포인트 집합을 비동기로 반환한다. 클라이언트가 보기를 다른 보기로 바꿀 때 이런 호출이 유용하다. Subscription은 이벤트 기반의 메커니즘으로써 사전 정의된 데이터 포인트 집합 내에 중대한 변화가 발생하면 서버가 클라이언트에게 통보한다.

2. OPC Server 개발

본 논문은 울진 표준형원전 GSE Simulator에 탑재되는 OPC server에 대한 자세한 Source 설명을 기술하였다. 프로그램은 Rockwell의 OPC Server Toolkit인 "RSI OPC/DDE Server Toolkit Library"를 이용하여 개발되었다. 대부분의 Rockwell code가 "C"함수이며 구체적인 C++ class를 제공하지 않으므로 이들을 별도로 개발하여 사용함으로서 효과적이고 체계적인 프로그램 개발을 추진하였다. 이 문서에서는 주로 GSE simulator와 관련된 부분을 집중적으로 기술하였으며 Microsoft MFC Class 활용과 같은 일반적으로 알려진 사항들은 생략하였다. 이 문서를 이해하기 위하여서는 다음과 같은 사전지식이 필수적이다.

- Microsoft Visual C++
- MFC MDI Framework
- OPC 기초지식
- Microsoft COM/DCOM 기초지식

2.1 Project 생성

Project는 다음의 option으로 생성되었다.

- MFC AppWizard (exe)
- Single document
- Document/View architecture support

2.2 Source file들

소스는 다음과 같은 파일들로 구성되어 있다.

- CommHdlr.h, CommHdlr.cpp
- Constant.h
- CsvFile.h, CsvFile.cpp
- DataFrame.h, DataFrame.cpp
- DataView.h, DataView.cpp
- DeviceMap.h, DeviceMap.cpp

- DeviceObject.h, DeviceObject.cpp
- Interface.h, Interface.cpp
- LogView.h, LogView.cpp
- MainFrm.h, MainFrm.cpp
- Stdafx.h, Stdafx.cpp
- Servercb.h, Servercb.cpp
- SystemArray.h, SystemArray.cpp
- SystemObject.h, SystemItem.cpp
- TimeoutDlg.h, TimeoutDlg.cpp
- OpcServDoc.h, OpcServDoc.cpp
- OpcServ.h, OpcServ.cpp

2.3 Server Interface

ProgID에 OPC Server List에 나타날 이름이 다음과 같이 지정되어 있다.

```
TCHAR szProductProgID[] = _T("Kepri.OpcServ.1");
```

2.4 통신관련 함수들

- CommHdlr.cpp
  - Data를 Read/Write하는 기능
- DeviceMap.cpp
  - Device기기를 포함하는 Map class
- DeviceObject.cpp
  - Device기기를 정의
- SystemArray.cpp
  - SystemItem을 포함하는 Array class
- SystemObject.cpp
  - 고정적인 MemoryObject

2.5 Source 개발

2.5.1 Library 자동으로 연결하기

Client에서 요청한 Tag가 Shared-Memory에 있는지 검색하기 위하여 필요한 함수들이 저장되어 있는 Library들을 자동으로 연결하기 위하여 다음과 같이 선언하였다. 다음과 같은 선언으로misc.lib, s3dll.lib, mstG.lib를 별도로 link에 정의하지 않아도 link시 자동으로 지정된다. 이때 mstG.lib는 현재 사용중인 시뮬레이터의 파일을 지정하여야 한다.

```
#pragma comment(lib, "misc.lib")
#pragma comment(lib, "s3dll.lib")
#pragma comment (lib, "mstG.lib")
```

2.5.2 mstG의 공유 memory 연결

mstG에서 확보된 공유 메모리는 C 함수로 만들어 졌으므로 다음과 같이 extern "C"를 추가한다. 이곳을 통하여 아래의 함수들이 시뮬레이터의 메모리에서 실시간 data를 읽고 쓰는 것이 가능하다.

```
extern "C"
{
DllImport int number_shared_globals;
DllImport STAB share_global_table[];
}
```

(가) 다음의 함수는 시뮬레이터에 정의된 전체 global의 개수를 가지고 온다.

```
int GetMaxSharedGlobals()
```

(나) 다음의 함수는 시뮬레이터에 정의된 특정 global의 정보를 인덱스를 이용하여 가지고 온다.

```
STAB* GetSharedGlobal(int nIndex)
```

(다) Mistic이라는 global table은 simulator의 다른 프로그램이 수행할 때 그 절대 address를 표시하므로 다음과 같이 그 이름을 가져와서 비교할 수 있게 하였다.

```
STAB* GetGlobalMistic()
```

(라) 모든 global table에서 GLOBAL_ 이라는 중복된 이름을 제거하고 보다 간단하게 client와의 통신이 가능하도록 다음과 같이 그 이름을 요약하는 함수를 제공한다. 예를 들어 'global_crs"를 szName에 인자로 넘기면 이 함수에서는 'global_crs"의 메모리상의 시작위

치정보 등을 가지고 있는 STAB 구조를 return 한다. 이 함수는 LookupPoint()에서 call 한다

STAB* GetSharedGlobal (LPCTSTR szName, bool bPackName)

2.5.3 공유 memory 초기화하기
공유메모리를 직접 access함으로 실시간으로 일어나는 상황을 감시가 가능하게 한다. 이것이 가능하게 하기 위하여 global table의 정보를 내부적 변수로 연결하는 code를 제공한다.

BOOL InitGlobalInfo(LPCTSTR szName, int nOffsetX, int nOffsetY, CSystemArray* pArray)

2.5.4 Static tag의2차원 array 지원
개발 초기버젼에서는 server에서 client에서 제공할 수 있는 tag를 별도의initial file에 저장을 하여 static하게 service하였으나 실제로는 client가 dynamic하게 tag를 생성하여 service할 필요가 생기고 또한 n-dimention array를 활용해야하므로 code가 수정되었다. 다음 code는 초기에 static한 tag를 access할 때 작성된 2차원 array지원 프로그램으로 사용되지 않으나 초기 code와의 호환을 고려하여 남겨놓았다. 현재 사용되는 array지원 code는 아랫부분의 array지원항목을 참조바란다.

void* GetMemoryOffsetPtr(void* pMemoryBase, int nOffsetX, int nOffsetY, int nDataSize, int nDimY)

2.5.5 공유 memory read
공유메모리 read는 읽고자하는 tag의 data type과 해당 tag의 시뮬레이터 공유메모리상의 메모리 위치값을 저장하고 있는 pValue를 이용하여 원하는 변수의 값을 read를 한다.

bool GetMemoryValue(DataType type, void* pValue, out object Value)

2.5.6 공유 memory write
공유메모리 write는 초기화된 memory pointer를 통하여 원하는 변수의 type을 casting하여write를 한다.

BOOL SetMemoryValue(const ICKITDATAVALUE& value, CSystemObject* pObject)

2.5.7 Tag의 memory 위치 검색
사용자가 개발하는 프로그램 중 핵심 모듈이다. 이 함수는 데이터 등록시 ParseItem() 함수에 의해 call 된다. 본 함수 내에서 시뮬레이터에서 그 환경에서 개발되지 않은 외부 프로그램을 위해 유일하게 제공하고 있는 "dbm_pid()" 함수를 내부에서 call 한다. szName과 nDim을 입력으로 하여 info 정보를 구한다. info는 szName의 type, szName이 소속된 global 명 등 시뮬레이터의 중요 정보를 가지고 있다.

void* LookupPoint(LPCTSTR szName, int nDim[], PIDINFO& info)

2.5.8 Tag의datatype 변환
GSE library는 1 char로 data type을 선언함에 반하여 Rockwell에서는 enum을 활용하여 tag의 data형태를 지정한다. 따라서 LookupPoint()에서 취득된 정보중에서 datatype을 변환하기 위하여 다음과 같은 routine이 제공된다.

int GetMemoryArrayType(char nType1, char nType2)

2.5.9 Tag의 n-dimension array 지원
시뮬레이터 변수들은 array를 사용하고 있으

며, 이것들을 OPC 표준과 연결하기 위하여 다음과 같은 내부규약을 만들었다.

차원	opc tag name	프로그램에서의 지정
기본	item	item
1차원	item#a	item(a)
2차원	item#a#b	item(a,b)
3차원	item#a#b#c	item(a,b,c)
4차원	item#a#b#c#d	item(a,b,c,d)

이렇게 함으로 client에서 일일이 dimension별로 고유한 이름의 tag를 생성해야하는 불편함을 제거하였으며 update의 효율성을 제고하였다. 그리고 이러한 변수들의 memory 위치를 찾기 위하여 다음과 같은 함수들이 제공된다.

CString ParseTagWithAddress(LPCTSTR szTagWithAddr, int nDim[])
void* GetOffsetWithDimention(char* pData, int nDataSize, int nFull[], int nDim[])

2.5.10 OPC Server 개발 현황
전력연구원은 개발된 OPC Server를 사용하면서 기능 보완의 필요성과 다른 프로젝트에서도 이를 이용하기 위해 다음과 같이 지속적으로 OPC Server를 개발하고 있다

적용과제	Server 명	Server Tool Kit	Client tool	비고
VRCATS-II	OpcServer	J&J	EON / Picasso	적용
고리1호기 시뮬레이터	OpcServer	J&J	Data Pant	적용
원전용 DCS 개발	OpcServer	Rockwell	DSIS	적용
Kins NPA	OpcServer KinsOpc	J&J Rockwell	Data Pant Data Pant	적용
고리1호기 시뮬레이터	GseServ	Softing	Data Pant	시험중
Kins NPA	KinsServ	Softing	Data Pant	시험중
고리1호기 시뮬레이터	GseServ	Iconics	Data Pant	계획중
Kins NPA	KinsServ	Iconics	Data Pant	계획중

4. 결론

전력연구원에서 개발하여 발전소에 공급한 시뮬레이터에서 OPC Toolkit을 이용하여 시뮬레이터와 외부 응용프로그램을 연동한 것은 본 과제가 처음이었다. 따라서 개발과정 중에는 시행착오도 많이 겪었다. 예를 들면, OPC Server가 시뮬레이터에서 원하는 값을 찾지 못하는 현상, Client에서 많은 변수를 요청하였을 때 초기 로딩에 많은 시간이 걸리는 문제, 원하는 시간내에 값을 Read/Write 하지 못하고 시간지연이 발생하는 현상, 변수 타입별 적절한 값을 불러오지 못하는 현상, 2차원 이상의 변수값을 읽을 수 없는 문제, 적절한 OPC Client를 구축하는 문제 등 많은 문제점들이 발생하였다. 현재는 이러한 문제점들을 해결하고 Client들에서 원하는 시간내에 원하는 성능을 발휘하도록 개발 완료되어 울진 표준형원전의 시뮬레이터 컴퓨터에 설치하여 운영하고 있다. 본 프로그램은 Windows 기반에서 사용하고 있는 GSE사의 SimSuit Power 시뮬레이터가 운영되고 있는 모든 시뮬레이터 환경에 사용 가능하다. 향후에는 OPC Server를 현재보다 많은 포인트수를 적용하여 초기의 로딩시간과 실시간성에 있어서의 성능을 보다 개선해 나갈 계획이다

참고문헌

[1] 박신열. "KSNP VRCATS용 OPC Server 개발", 기술보고서, 전력연구원, March 2002.
[2] CERN, "OPC (OLE for Process Control)", CERN, Joint Controls Project
[3] Thomas J. Burke, "The Performance and Throughput of OPC", A Rockwell Software Perspective, June 11, 1998