

# 실시간 자바를 이용한 HVDC Control and Protection 시스템 구현에 관한 고찰

서민욱

LS 산전(주)

## The study for implementation of HVDC Control and Protection system using realtime JAVA.

M.W. Sua

LSIS

### ABSTRACT

본 논문에서는 실시간 자바를 사용하여 초고압 송전 설비의 제어 및 보호 장치의 구현에 대해 고찰 하고자 한다. 초고압 송전 설비는 대단위 시스템 이라서 객체 지향 프로그래밍이 유리하지만, 실시간 운영체제 상에서 고속으로 실시간 동작을 해야하므로 객체지향 프로그래밍이 모순 된다 할 수 있다. 실시간 자바를 통해서 대형 HVDC 시스템에서 객체 지향 설계의 유연함과 고속 실시간 시스템을 구현 할 수 있는 장점을 동시에 가질 수 있는 방법에 대해 고찰 해 보자.

### 1. 서 론

이미 상용으로 사용할 수 있는 실시간 자바 가상 머신이 판매되고 있다. 이러한 제품 가운데에는 특정 RTOS를 사용하는 임베디드 시스템에서도 사용할 수 있는 종류도 존재한다. 기 개발된 자바 가상 머신 들을 이용하면 검증된 제품을 바로 시스템에 적용할 수 있으므로 개발 기간 단축 및 데이터 처리 및 통신을 위한 코드 개발이 용이한 장점이 있다. HVDC 제어기는 대 단위 임베디드 시스템인 경우가 대부분 이기 때문에 객체 지향 프로그래밍 언어를 도입하는 것이 유리하다. 요구 사항이 변경 되더라도 유연하게 대처 할 수 있는 언어는 객체 지향 언어가 유리하다. 대부분의 경우 임베디드 시스템의 경우 C 언어를 많이 사용하지만, C++를 사용 하더라도 객체 지향성은 첨가 할 수 있으나, 운영 체제 혹은 하드웨어가 변경 되면, 소스 코드 변경은 필연 적이다. 실시간 자바는 실시간 성을 보장 하면서도 객체 지향성 확보 및 운영 체제, 하드웨어 변경 시에도 소스 코드 변경이

필요 없다. 이러한 제품으로 데이터 HVDC 제어기를 구현하는 형태는 본문에서 소개하도록 한다.

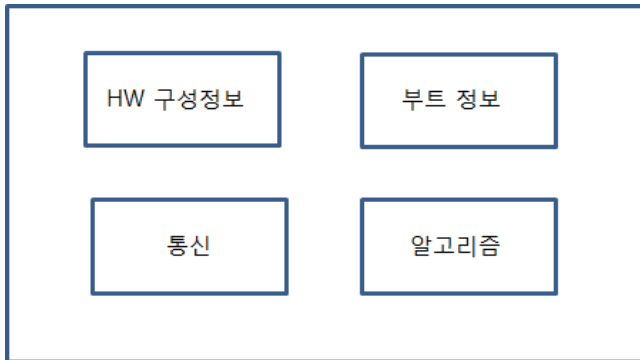
### 2. 본 론

#### 2.1 실시간 자바의 구성

일반적으로 실시간 소프트웨어 들의 특징은, 정해진 시간 안에 시그널에 대한 응답을 보장하고, 가비지 컬렉션을 한다면, 실행 중지됨이 없어야 한다. 또한 우선 순위 역전 등의 실행 불확실성을 배제할 수 있어야 한다. 이러한 제약 사항에도 불구하고, 객체 지향성을 코드에 붙여넣는 것은 상당히 고무적인 일이다. 일반적으로 객체 지향 언어들 특히 자바는 가비지 컬렉션과 컴파일 결과물이 실시간 실행 성에 가장 걸림돌이 되므로 이 두가지를 해결하면, 훌륭한 실시간 소프트웨어를 작성할 수 있는 언어로 둔갑한다.

자바 런타임은 JIT(Just In Time) 컴파일러를 사용하여 자바 어플리케이션 에서 가장 자주 실행되는 메소드 들을 네이티브 코드 환경에서 동적으로 생성한다. 실시간 환경에서 이러한 속성은 특정 시간 내에 동작해야 한다는 제약 사항을 만족할 수 없다. 이러한 제약 사항을 극복하기 위해 각 실시간 자바 가상 머신 제조 업체 에서는 AOT(Ahead of Time) 컴파일 방식을 사용하는 자바 가상 머신을 제조하여 이러한 한계를 극복한다. AOT 컴파일은 어플리케이션이 실행되기 전에 어플리케이션의 자바 메소드용 네이티브를 생성한다. 이렇게 함으로써, 동적 컴파일로 인한 비 결정론을 극복하고, 컴파일에 의한 성능 향상도 기대할 수 있다. 가비지 컬렉션 또한 특정 객체를 특정 시간 까지 가비지 컬렉션 할 수 있어야 메모리에 대한 관리가 해결 될 수 있다.

## 2.2 제어 및 보호 시스템에서 실시간 자바의 유연성



〈그림〉HVDC 제어 및 보호 시스템 SW의 구성

〈그림〉에서는 HVDC 제어 및 보호 시스템 SW의 구성을 보여준다. H/W 설정, 부트 정보, 통신, 제어 알고리즘, 등으로 나누어 진다. H/W 설정 및 부트 정보에 관련해서는 객체 지향이 아니더라도 충분히 해결할 수 있다고 생각할 수도 있지만, H/W 경우를 살펴 보면, 보드가 변경되거나, 그에 따른 부트 정보가 변경되었을 때, 이러한 것을 객체 지향화 하여 설정해 놓고, 공통 클래스에서 상속 받는 방식으로 어플리케이션을 구성해 놓으면, 코딩을 변경할 필요 없이 다양한 보드에 대한 인스턴스를 자동으로 선택하여 실행 하도록 구성할 수 있다. 이렇게 하면 코드량의 변경 및 수행 속도에 영향을 줄 가능성은 있지만 H/W 상황에 따른 유연성을 제공하여, 뛰어난 유지 보수성을 제공한다. 통신의 경우를 살펴보면 초기 시스템을 구축할 시에는 LAN 통신만을 이용하여 시스템을 구축하였으나, 시간에 지남에 따라, 개발 및 유지 보수 도중, 광통신, 필드 버스 등 다양한 통신 추가에 대해서도, 클래스 추가 및 인스턴스 생성으로 다양한 상황에 대응할 수 있도록 할 수 있다. 알고리즘의 경우에 있어서도 마찬가지로의 장점을 제공한다. 알고리즘을 카테고리 별로 상황을 분류하여, 클래스 화 하여, 유지 보수 한다면, 인스턴스의 추가 삭제 등의 간단한 변경으로 다양한 상황에 대처할 수 있도록 코드를 유지 보수 할 수 있다.

또 다른 실 시간 자바의 장점으로는 포인터를 사용하지 않아도 되는 사실이다. C언어로 작성하는 대부분의 임베디드 프로그램은 메모리 관리라는 공통적이고도 잠재적인 메모리 관리 문제를 알고 있다. 포인터를 할당하고 해제하는 과정 중에서 프로그램의 구조적인 문제이건, 프로그래머의 실수 이건, 여러 문제를 나타나게 된다. 이렇게 프로그램이 작성된 경우 널 포인터를 메모리에 할당 해서 프로그램의 런타임 오류를 나타낸다든지, 할당 메모리의 경계를 초과하여 오류를 발생 시킬 수도 있으며, 심지어 메모리 누수 현상을 일으켜서, 지속적으로 시스템의 메모리를 잠식하기도 한다. 실시간 자바는 이러한 문제의 해

결책 중의 하나이다. 객체의 인스턴스를 자동으로 힙메모리 상에 할당 하지만 가비지 콜렉션을 통하여 메모리 관리를 거의 자동으로 해결해 주므로 프로그램 작성자가 일일이 포인터에 관련하여 관리하지 않아도 되도록 한다. 이러한 것이 실시간 시스템에서도 가능한 것은 실시간 자바의 경우 여타 일반 자바 가상 머신과는 달리 가비지 콜렉션 마저도 특정 시간안에 해결하는 알고리즘을 가지고 있기 때문에 가능한 것이다. 또한 이러한 것들이 실시간성으로 동작하기 위해서는 하위 운영체제가 실시간 운영체제 이어야 함은 필수적이다.

임베디드 시스템이 점점 대형화 되면서 분업화, 단위 작업화는 필수적이 되어 버렸다. 실시간 자바를 이용하면, 초고압 송전 설비의 제어기처럼 큰 규모의 소스 코드를 제작하고, 관리함에 있어서, 클래스와 같은 특정 단위별 분업화 및 단위 시험, 형상관리, 코드 인스펙션 등 소프트웨어 공학적인 측면의 관리를 좀더 효율적으로 수행하는데 있어 편리한 틀을 제공한다. 이러한 것을 통하여 개발 조직의 구성 및 운영을 더욱 편리하게 하는 장점을 지니게 된다.

## 3. 결 론

본 논문에서는 초고압 송전 설비 제작 시 제어 및 보호 장치 SW에 객체 지향을 도입할 경우의 장점들을 살펴 보았다. 제시된 방법들 중 진행 중인 프로젝트에 대하여 적절한 가격대 성능 사양을 고려한 후 적용한다면 향후 개발 기간 단축 및 구축 후 변경 및 유지 보수에 대한 유연성을 갖춘 제어 및 보호 장치를 개발 할 수 있다고 생각한다.

## 참 고 문 헌

- [1] <http://www.aicas.com> : 아이카스
- [2] 자바 튜토리얼 : 성안당