Sequence Diagram을 이용한 안전등급 PLC 운영체제의 인터페이스 설계

이영준[0] 권기춘 이장수 김장열 차경호 천세우, 손한성*
한국원자력연구소, *㈜액트
{yjlee426, kckwon, jslee, jykim, khcha, swcheon}@kaeri.re.kr, *hsson@actbest.com

A Design of the Operating System Interface for Programmable Logic Controller Using Sequence Diagram

Young-Jun Lee[0] Kee-Choon Kwon Jang-Soo Lee Jang-Yeol Kim
Kyung-Ho Cha Se-Woo Cheon, *Han-Seong Son
Korea Atomic Energy Research Institute, *Atomic Creative Technology.

요 약

본 논문은 Sequence Diagram을 이용한 안전등급 PLC(Safety-Grade Programmable Logic Controller) 운영체제의 인터페이스 설계명세를 기술한다. 원자력발전소에 사용하기 위한 안전 소프트웨어(Safety Software)의 규제기준인 Reg. Guide 는 IEEE Std 1016, IEEE Std 1016.1 의 설계명세서 작성표준에 따라 작성하도록 요구하며, 이러한 규제기준과 기술표준을 안전등급 PLC를 위한 운영체제 소프트웨어의 설계명세서도 만족해야 한다. 특히 운영체제와 외부 장치들 사이의 인터페이스를 위해 실시간 특성의 표현에 장점을 갖는 Sequence Diagram을 적용함으로써 운영체제의 인터페이스에 대한 정확성, 완전성, 그리고 일관성을 향상시킬 수 있었다.

1. 서론

원자력 발전소에 사용하기 위한 안전등급 제어기기의 소프트웨어는 미국 NRC(Nuclear Regulatory Commission)의 Regulatory Guide(이하 Reg. Guide)에 따라 개발된다. 이는 안전성이 최우선으로 보장되어야 하는 시스템의 특성상 개발 초기부터 소프트웨어에 대한 품질을 높일 수 있는 방법을 요구하기 때문이다. NRC의 Reg. Guide 1.173[1] 에서는 산업표준으로 사용하고 있는 IEEE. Std 1074[2]을 승인하고 있고 이 표준에서 참조하고 있는 IEEE. Std 1016[3], IEEE Std 1016.1[4]에 따라 설계명세서를 작성하여 설계단계의 품질을 보장하게 된다. KNICS(Korea Nuclear Instrumentation & Control Systems) 과제에서 개발되는 안전등급 제어기기(Safety-Grade Programmable Logic Controller)의 운영체제는 설계명세서를 작성할 때 내,외부 장치들과의 인터페이스 표현을 위해 Sequence Diagram을 사용하였다. 본 논문은 Sequence Diagram을 통해 인터페이스가 표현되고 있는 사항을 기술한다.

2. 관련연구

2.1 소프트웨어 설계명세서

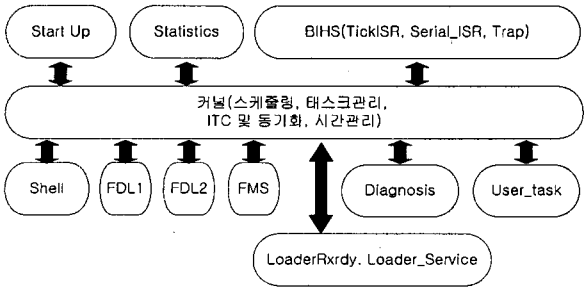
소프트웨어의 개발생명주기 중 설계단계는 요구사항단계 이후에 수행되는 단계로써 요구사항에 맞게 소프트웨어의 구조를 파악하고 나아가 구현을 위한 발판을 마련하는 것으로, 생성하여야 하는 것은 소프트웨어 설계명세서이다. 이 명세서에는 소프트웨어가 가져야 하는 설계개체, 개체들의 속성들이 기술되고 설계개체들의 구조와 그들 사이의 연관관계, 그리고 설계개체들간에 주고 받는 데이터에 대해 기술하여야 한다.

2.2 설계명세서 작성을 위한 Sequence Diagram

UML(Unified Modeling Language)는 소프트웨어 시스템을 분석하고, 시각적으로 표현하고, 설계하고, 문서화하기 위해서 사용하는 범용의 모델링 언어이다[5]. UML에서 시스템을 모델링할 수 있는 방법은 class diagram, sequence diagram, state diagram, activity diagram 과 같이 여러 종류가 있는 데 표현하고자 하는 특성에 따라 서로 다른 표현 방법을 사용하여 시스템을 모델링한다. 그 중 시스템 설계의 동적인 행위들을 표현하기 위해서는 sequence diagram을 많이 사용하게 된다. 이는 sequence diagram이 시간개념이 포함되어 동작하는 모듈들의 상호행위들을 표현할 수 있기 때문이다[5]. 또한 sequence diagram을 이용하여 개체들간에 전달되는 메시지와 전달 시간을 파악할 수 있다.

3. 안전등급 PLC 운영체제 설계 구조

안전등급 PLC 운영체제의 설계 개체는 [그림 1] 에서 보는 것과 같이 여러 개의 태스크들이 서로 독립적인 기능을 수행하며 동작한다. 운영체제는 일반적인 실시간 운영체제의 핵심이라고 할 수 있는 커널(Kernel) 부분과 다른 외부장치 및 커널과의 인터페이스를 담당하는 시스템 태스크(System Task)로 나눌 수 있다.



[그림1] 안전등급 PLC 운영체제의 설계개체

각각의 태스크들은 커널의 스케줄링 기능에 의해 시간주기를 가지고 실행되며 운영체제의 동작을 수행한다.

4. 안전등급 PLC 운영체제의 설계명세서

안전등급 PLC 운영체제의 설계명세서는 안전등급 PLC 소프트웨어 설계명세 작성절차서[6] 에 따라 작성되었다. 그 작성목차는 [그림 2] 와 같다.

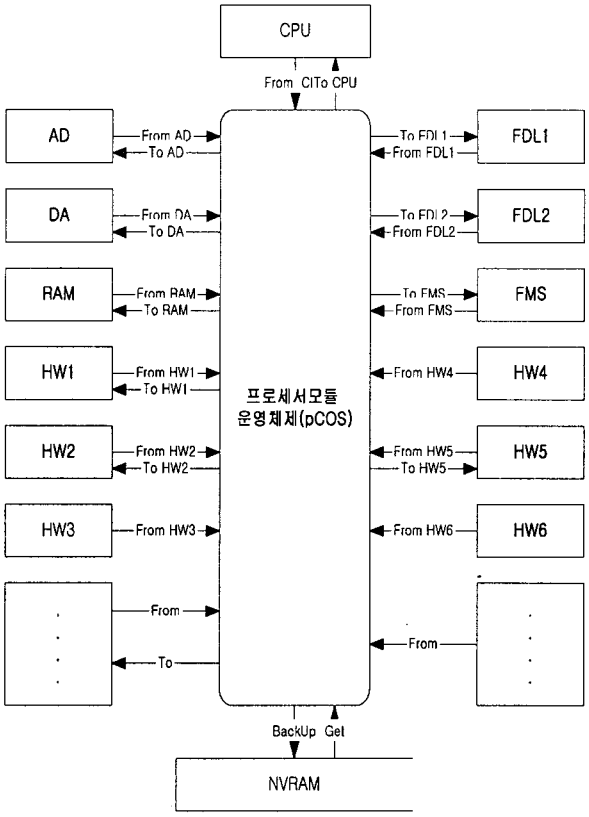
1. 목적
2. 범위
3. 참고문헌
4. 용어정의 및 약어
5. 안전등급 PLC 운영체제 설계개요
   - A. 안전등급 PLC 운영체제 설계개체
   - B. 안전등급 PLC 운영체제 설계속성
6. 안전등급 PLC 운영체제
   - A. 분해 설계
   - B. 의존성 설계
   - C. 인터페이스 설계
   - D. 상세설계

[그림 2] 안전등급 PLC 운영체제 설계명세서 목차

이는 IEEE Std. 1016 에서 추천하는 설계명세서 작성형식과 유사하다. 서론에서 설명하였듯이 NRC. Reg Guide 에서 IEEE Std 문서를 규제요건으로 승인해서 사용하기 때문이다. 5장에서는 운영체제를 구성하고 있는 설계개체들을 정의하고 그 개체의 속성에 대해서 기술한다. 안전등급 PLC의 개체는 StartUP 소프트웨어와 8개의 시스템 태스크, 그리고 커널로 나누어 분할될 수 있다. 6장의 분해설계에서는 설계개체들이 소프트웨어의 구조에 영향을 주는 정도까지 모듈들을 분해하여 표현하고, 의존성 설계에서는 설계개체들간의 관계들을 파악한다. 개체들간에 서로 주고 받는 메시지나 데이터들이 무엇인지 파악하기 위하여 인터페이스 구조설계를 하여야 한다. 상세설계는 시스템의 구현을 위해 시스템 기능들에 대해서 자세하게 묘사하는 기술이다.

5. 인터페이스 설계를 위한 Sequence Diagram

안전등급 PLC 운영체제는 커널과 시스템 태스크, 시스템 태스크와 시스템 태스크간 상호동작하며 운영된다. 태스크들이 동작하기 위해서는 커널을 통해 CPU의 사용권한을 얻고 주어진 시간 내에 다른 태스크들과 서로 인터페이스를 하며 역할을 수행하게 된다. 시스템 태스크들이 커널의 서비스를 받으며 동작하는 것은 의존성 설계기술로 표현하고 운영체제와 하드웨어들과의 데이터에 대한 관계는 인터페이스 표현을 통해 설명될 수 있다.

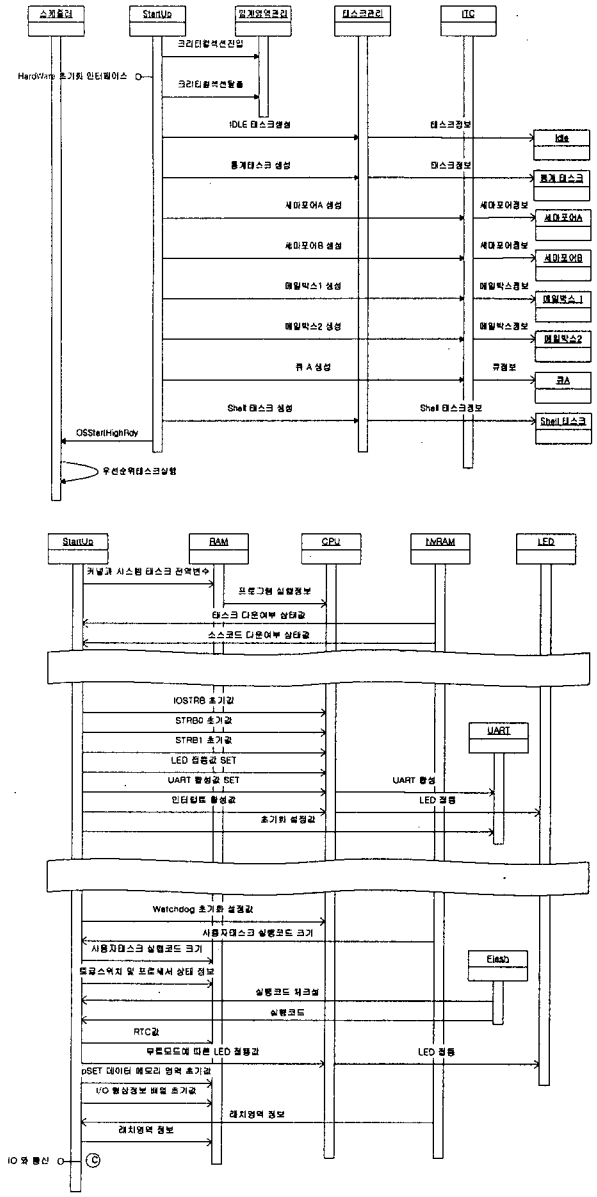


[그림 3] 안전등급 PLC 운영체제 인터페이스 개체

안전등급 PLC 운영체제의 인터페이스를 표현하기 위해서 Sequence Diagram 을 사용하였는데 Sequence Diagram을 채택한 이유는 두 가지이다.

첫째, 설계명세서상에 정의된 인터페이스는 개체들간에 주고받는 메시지, Signal, 데이터들이다. 이러한 메시지를 송수신하는데 가장 많이 이용하는 모델링 방법이 바로 Sequence Diagram 이기 때문이다.

둘째, 기술하려고 하는 인터페이스는 단순한 모듈들간의 데이터 송수신이 아니라 바로 운영체제와의 송수신이기 때문에 시간적인 개념과 동기화 기술이 포함되어 있다. 여러 다른 모델링 기법 중 Sequence Diagram은 연속적인 시간 사이에 나열되어 있는 모듈 사이의 상호동작을 보여주게 된다[7].



[그림 4] 인터페이스 Sequence Diagram

위의 그림은 안전등급 PLC 운영체제의 초기화 부분에 대한 인터페이스를 나타내고 있다. 커널이 가지고 있는 여러 서비스들을 각각 독립적인 개체로 두고, 초기화 때 생성되는 태스크와 동기화 리소스들의 생성절차를 표현한다. 커널서비스를 논리적으로 나누어서 각각의 개체로 표현한 것은 태스크나 동기화 리소스들이 생성될 때 초기화 값이 매개변수로 전달되기 때문이다. 즉 태스크를 생성할 때는 운영체제의 초기화 부분에서 커널의 태스크관리 서비스에 생성명령을 송신한다. 이 명령을 수신한 태스크관리 서비스는 태스크 생성을 위해 필요한 정보를 가지고 실제 수행되는 태스크를 생성하게 된다. 이 때 태스크 정보는 태스크 마다 서로 다른 데 이에 대한 명세는 자연어를 통해 기술하거나 상세설계에서 세부적으로 표현한다. 운영체제에서 사용하는 하드웨어들을 개체로 정의하고 이 개체들과 서로 주고 받는 메시지나 신호, 명령들을 표현하기 위해서 Sequence Diagram을 사용한다. 통신이나 입출력장치들과 같은 외부 하드웨어와의 인터페이스는 공유메모리를 통해 서로 데이터 송수신을 하기 때문에 운영체제는 이러한 공유메모리의 데이터를 읽고 쓰는 일을 담당하게 되는 데, 이 때 발생되는 데이터의 이동을 개체들간에 서로 표현하게 된다.

Sequence Diagram을 사용하여 인터페이스를 표현함으로써 개체들간의 데이터 일치성을 확인할 수 있고, 커널자원을 획득하는 순간과 인터페이스 시점에 대한 시간적인 파악이 가능하여 서로 독립적으로 수행되는 멀티태스킹의 동기파악이 수월해졌음을 알 수 있다.

6. 결론 및 향후 계획

안전등급 PLC의 운영체제의 설계명세서를 작성하여 소프트웨어의 구조와 설계개체들의 속성, 그리고 개체들간의 관계와 인터페이스들을 표현하기 위해 UML 모델링 방법중의 하나인 Sequence Diagram을 사용하였다. 이를 통해 운영체제의 시간적인 표현이 가능하여 운영체제의 동작을 쉽게 이해할 수 있고, 운영체제와 다른 하드웨어와의 인터페이스를 표현하여 하드웨어들과의 송수신 메시지에 대한 파악이 더욱 쉬워졌다. 또한 운영체제의 커널 서비스와 하드웨어와의 인터페이스가 이루어지는 시점과 시간을 표현할 수 있어 운영체제의 동기화에 대한 분석이 더욱 용이해졌음을 확인할 수 있게 되었다. 추후 Sequence Diagram의 시뮬레이션이 가능한 도구를 통해 운영체제가 동작하는 과정을 미리 파악할 수 있게 할 계획이다. 이를 통해 실시간으로 동작하는 운영체제의 성능을 향상시킬 수 있을 것이고 인터페이스 동작을 수행할 때 발생되는 오류를 파악하여 안전등급 PLC 의 안전성을 더욱 향상시킬 수 있을 것이다.

참고문헌

[1] USNRC Reg. Guide 1.173, Development of Software Life Cycle Processes for Digital Computer Software Used in Safety Systems of Nuclear Power Plant, 1997.

[2] IEEE Std. 1074-1997, Standard for Developing Software Life Cycle Processes.

[3] IEEE Std. 1016-1998, Recommended Practice for Software Design Descriptions.

[4] IEEE Std. 1016.1-1993, Guide to Software Design Descriptions.

[5] G.Booch, J.Rumbaugh, and I.Jacobson. "The Unified Modeling Language User Guide". Addison-Wesley, 1999.

[6] "안전등급 PLC 소프트웨어 설계명세 작성절차서", 한국원자력연구소, 2004.

[7] Xiaoshan Li, Zhiming Liu, and He Jifeng, "A Formal Semantics of UML Sequence Diaram", Proceeding of the 2004 ASWEC

[8] OMG, "Unified Modeling Language specification, version 2.0". Object Management Group. http://www.omg.org/technology/documents/.2003