# 확장 Shlaer-Mellow 방법 기반의 임베디드 소프트웨어 아키텍처 프레임워크 개발 방법 연구

오광근[O*], 김종배*, 문전일*, 박수용**
LG산전 연구소*, 서강대학교 컴퓨터학과**
{kkoh[O], jimoon, jongbaek,}@lgis.com, sypark@mail.sogang.ac.kr

# A study on the Development Method for the embedded software architecture framework with the extended shlaer-Mellow method

KwangKeun Oh[O], JongBae Kim, JeonIL Moon, SooYong Park
LG Industrial Systems R&D Center, Sogang University

**요 약**

동일 제품군에 대한 소프트웨어 재사용 요구 와 디지털 복합 제품군의 등장으로 임베디드 소프트웨어에 대한 아키텍처 중요성은 날로 증가하는 추세에 있다. 하지만 임베디드 시스템 개발이 하드웨어 중심으로 이루어지는 특성상 소프트웨어 아키텍처에 대한 연구는 미비한 현실이다. 이에 본 연구에서는 임베디드 시스템의 핵심 아키텍처 요소를 서비스로 보고, 서비스 제공을 위해 기능 중심의 아키텍처 스타일 및 프레임워크 개발 방법을 제시하고 인버터 제품에 대한 사례연구를 통해 임베디드 시스템 개발에 효과적임을 확인하였다.

## 1.서론

임베디드 시스템의 복잡도는 급속히 발전하는 전자 정보 통신 기술 발달과 함께 활동 영역도 날로 증가하는 있다. 또한 간단한 전자회로만으로 동작하던 제품들에 대한 고기능 및 네트워크 요구도 증가하고 있으며, 제품 라이프 사이클도 점점 짧아지는 추세에 있다. 이런 변화들로 인해 소프트웨어 비중은 점점 증가하고 있으며, 기존 하드웨어 중심의 개발체제로는 이와 같은 산업 변화를 따라갈 수 없게 되었다. 결국 임베디드 시스템 개발에도 생산성과 신뢰성을 동시에 만족하는 소프트웨어 개발 패러다임 요구가 늘어가고 있으며, 아키텍처 기반의 소프트웨어 개발 방법이 그 대안으로 제시되고 있다.

이에 본 연구에서는 임베디드 시스템 개발 시, 디자인 수준에서 사용 가능한 기능 중심의 아키텍처 스타일과 아키텍처 프레임워크 개발 방법을 제시하고자 한다. 연구를 통해 도출된 기능 중심의 아키텍처 스타일은 OMG(Object Management Group)의 UML 2.0표기인 Structure Diagram을 사용하여 컴포넌트와 커넥터로 표현하였으며, LG산전 임베디드 제품인 인버터를 통한 사례 연구로 그 적용 가능성을 확인하였다.

## 2. 임베디드 시스템에서의 소프트웨어 아키텍처 요소

임베디드 시스템이란 내장형 시스템이란 용어로도 사용되며, 다른 시스템에 의존하지 않고 내부 마이크로 프로세서를 통해 하드웨어를 제어하며, 제공하는 서비스를 위해 특정 작업과 기능만이 수행되도록 설계되어 있다. 즉, 임베디드 시스템은 그 사용 용도가 설치되는 프로그램에 의해 좌우되는 것이 아니라 처음부터 특정 목적을 위해 전문적인 용도로 활용 되고 있다[1]. 이와 같은 서비스 특성을 고려한 소프트웨어 개발을 위해서는 기능관점의 아키텍처 개발이 이루어져야 한다. 또한 대부분의 임베디드 시스템은 정해진 시간에 원하는 서비스를 제공해야 하는 실시간 특성을 가지고 있으므로, 실시간성에 의한 아키텍처 스타일 분류도 중요한 아키텍처 요소이다. 서비스와 실시간성 이외에 성능,보안,신뢰성등도 아키텍처를 구성하는 중요한 품질 요소로 활용될 수 있다.

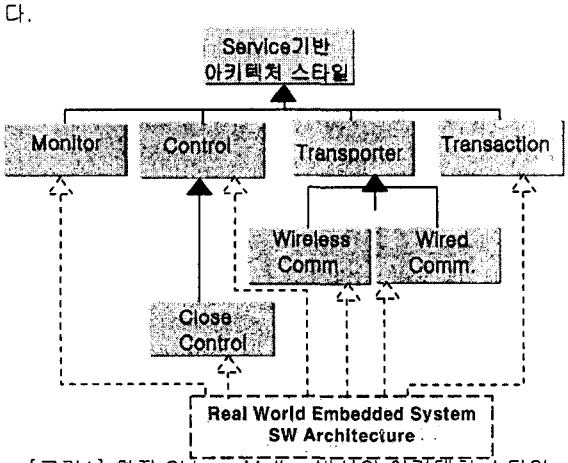
## 3. 기능 중심의 임베디드 소프트웨어 아키텍처 프레임워크 개발 방법

아키텍처를 구성하는 여러 요소 중, 본 연구에서는 임베디드 시스템 구조를 결정짓는 요소로 서비스 특성으로 선정하였다. 왜냐하면 성능,실시간성,보안,신뢰성 등의 아키텍처 요소는 기본적인 기능 특성 하에서 도출 될 수 있으며, 특히 시스템 개발 초기에 아키텍처 프레임 워크 생산을 위해서는 기능관점의 아키텍처 개발 방법 도입이 우선되어야 한다고 생각한다. 이에 기능 중심의 임베디드 시스템 소프트웨어 아키텍처 연구 방향을 다음과 같이 설정하였다.

첫째, 임베디드 시스템 개발 시, 시스템 구조에 결정적인 영향을 주는 기능중심의 아키텍처 스타일 도출
둘째, 아키텍처 스타일 구성을 컴포넌트 와 커넥터의 연결 관계로 묘사
셋째, 아키텍처 프레임워크 생성을 위한 방법론 제시

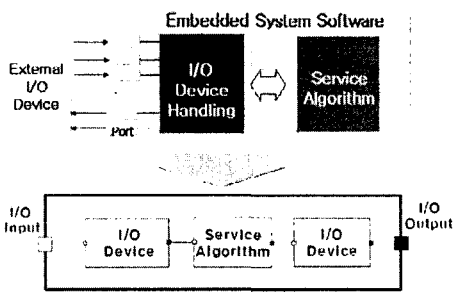
먼저 아키텍처 스타일 연구는 임베디드 소프트웨어 아키

텍처 특성을 기능 관점에서 Monitor and Control, Transporters, Transactions의 3가지로 분류한 Shlare Mellow 방법을 기반으로 연구를 수행하였다[2]. 제시된 Shlare Mellow방법은 전체 임베디드 시스템 서비스에 대한 분류기준을 제시하는데 큰 의미를 둘 수 있으나 디자인 수준의 아키텍처를 도출하기에는 분류를 세분화 할 필요가 있다. 또한 설명 위주의 아키텍처 스타일 묘사로는 디자인 수준의 컴포넌트를 도출하는데 한계가 있다.
이에 본 연구에서는 기 제시한 추상적인 아키텍처 요소를 보완하기 위해 아키텍처 스타일을 임베디드 시스템의 구조 특성에 따라 I/O 디바이스 처리 관점과 서비스 알고리즘 관점에서 그림1과 같인 7개의 스타일로 세분화하여 다.



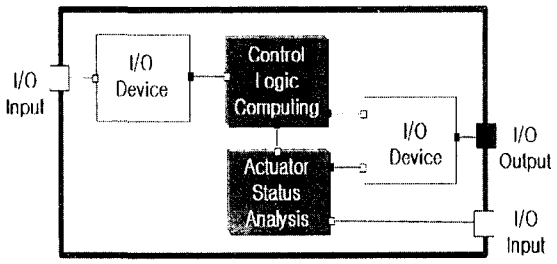

[그림1] 확장 Shlare-Mellow방식의 아키텍처 스타일

그림1에 나타난 7개의 아키텍처 스타일은 임베디드 시스템에 독립적으로 적용될 수도 있으며, 여러 개를 상속받아 복합적으로 적용될 수도 있다. 각각의 아키텍처 스타일은 컴포넌트와 커넥터의 관계로 정의하여 UML 2.0 (Proposal Version)표기법인 structure chart를 사용하여 나타내었다. UML2.0 은 실시간 임베디드 특성을 고려한 모델링 기법이며, 특히 structure chart는 아키텍처 구성 요소인 컴포넌트와 커넥터를 나타낼 수 있는 아키텍처 모델링 기법이다[3].



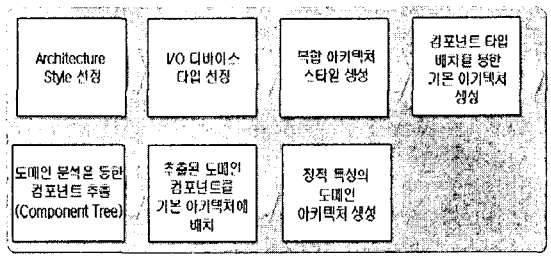

[그림2] 기능 중심의 아키텍처 스타일 구성 요소

그림2는 임베디드 시스템의 구조적인 특성 요인인 I/O 디바이스 처리와 서비스 알고리즘을 기준으로 제시된 임베디드 시스템 아키텍처 스타일을 구성하는 방법을 나타낸 것이다. 즉, 아키텍처 스타일은 I/O디바이스 처리 컴포넌트 타입과 제어 알고리즘 컴포넌트 타입으로 나누어진다.




[그림3] ' Close Control' 아키텍처 스타일

그림3은 Close Control 아키텍처 스타일을 나타낸 것이며, I/O Driver Input Handler에서 지령 입력을 받아 Control Logic Computing을 통해 제어 연산을 수행 후, Display Driver Handler와 Actuator Driver Handler를 통해 제어 신호를 내보내는 역할을 수행한다. 또한 Close Control제어를 위한 Actuator Status Analysis컴포넌트를 두어 Actuator의 현재 값을 읽어와 내부 파라미터값으로 변화하는 역할을 담당한다.
이렇게 제시된 아키텍처 스타일과 각 스타일의 구성요소인 컴포넌트와 커넥터 관계로부터 제품 개발 시 사용할 아키텍처 프레임워크 개발 방법을 아래 그림4와 같이 제시 하였다.



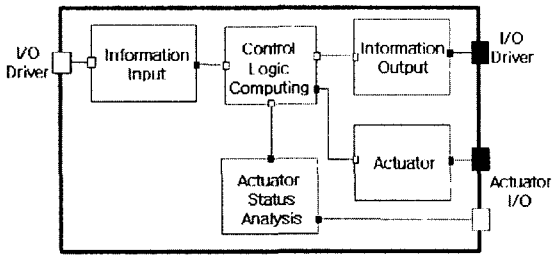

[그림4] 기능중심의 아키텍처 프레임워크 개발 방법

먼저 기능 중심의 아키텍처 스타일을 중 도메인이 제공해야 하는 서비스 특성을 고려하여 아키텍처 스타일들을 선택하고, 시스템이 제공하여야 하는 I/O디바이스 타입에 따라 컴포넌트 타입을 선정하여 복합 아키텍처 스타일을 도출한다. 이렇게 도출된 아키텍처 스타일에 기본 컴포넌트 타입을 배치하고, Component Tree를 이용해 도메인에 사용할 컴포넌트들을 추출한다. 이렇게 추출한 컴포넌트들을 기준으로 도메인에 특화된 아키텍처 프레임워크이 생성된다.

## 4. 사례 연구

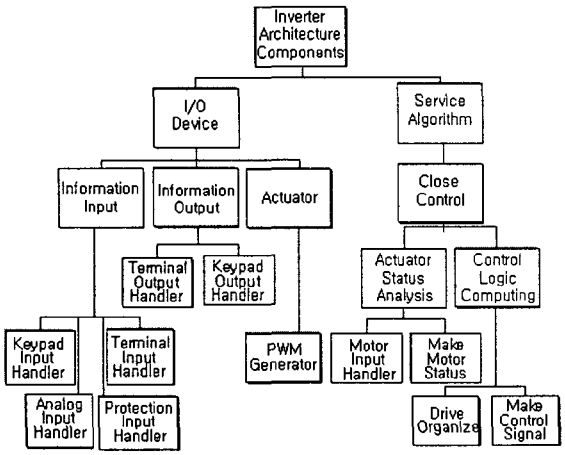
제시된 기능 중심의 아키텍처 프레임워크 개발 방법이 실제 임베디드 시스템 개발 환경에 적용 가능한지를 확인고자 산업용 임베디드 시스템인 인버터 제품에 대한 사례 연구를 수행하였다. 인버터는 외부로부터 입력 주파수 신

호 및 입력 지령를 받아 내부 계산을 통해 출력 PWM신호를 출력 서비스를 제공하는,전형적인 ' Close Control 아키텍처 스타일 타입이다. Close Control 아키텍처 스타일 타입에 I/O 디바이스 컴포넌트 타입은 Information Input, Information Output, Actuator로 분류할 수 있다. 선택된 아키텍처 스타일에 I/O디바이스 타입을 고려하면 그림5와 같은 기본 아키텍처가 생성된다.



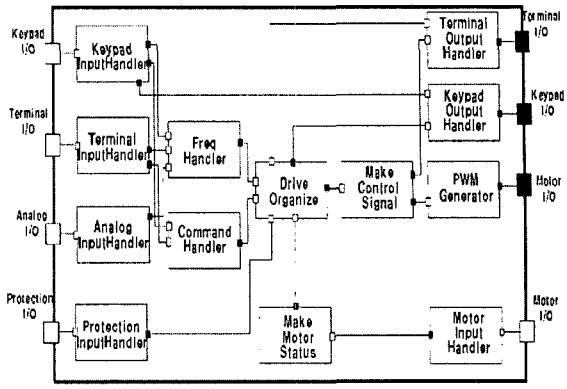

[그림5]컴포넌트 타입들을 배치한 인버터 기본 아키텍처

인버터 도메인에 특화된 컴포넌트들을 추출하기 위해 기본 아키텍처 컴포넌트 타입들을 기반으로 Component Tree분석을 실시한다.




[그림6] 기능 중심의 인버터 아키텍처 스타일에 대한 Component Tree

그림6는 각각의 컴포넌트 타입들에 해당하는 컴포넌트들을 도메인 특성에 맞게 도출한 후 컴포넌트 타입들과 연결한 Component Tree를 나타낸 것이다. 이렇게 도출한 도메인 컴포넌트들을 기본 아키텍처를 근간으로 배치하면 그림7과 같은 아키텍처 프레임워크가 생성된다. 생성된 아키텍처 프레임워크는 기능중심의 인버터 소프트웨어 개발 기준이 된다. 제공된 컴포넌트들은 독립적인 Subsystem으로 재사용단위가 되며, 각 컴포넌트들간의 통신은 Port를 통해 이루어진다.
이렇게 생성된 아키텍처 프레임워크는 기존 아키텍처가 존재하지 않는 소프트웨어 구조보다 각 컴포넌트들간의 독립성이 보장되며, 컴포넌트들간의 정보흐름 파악이




[그림7] 기능중심의 인버터 아키텍처 프레임워크

쉬운 구조임을 확인하였으며, 시스템 추상화 단위를 계층 구조로 관리할 수 있어서 소프트웨어 복잡성 관리에 좋은 방법이라는 것을 확인 하였다.

## 5.결론 및 향후 연구

임베디드 시스템 아키텍처에 대한 연구는 다양한 도메인 특성상 일반화 시키기가 그리 쉬운 작업이 아니다. 또한 하드웨어 중심으로 개발이 수행되고 있어 소프트웨어 관점의 아키텍처 요소 추출이 쉽지 않은 현실이다. 하지만 동일한 아키텍처를 가진 시리즈 제품 군에 개발 활동이 점차 늘어가는 추세에 따라 소프트웨어 재사용을 높이고 디자인 활동을 용이하기 위해서 아키텍처 중심의 개발 요구가 자연스럽게 추진되고 있다. 이에 본 연구에서는 임베디드 시스템 아키텍처 스타일들 서비스 제공 단위로 제시하고, 각 스타일과 I/O디바이스 타입에 의해 기능 중심의 아키텍처 프레임워크를 개발 할 수 있는 방법을 제시하였다. 또한 사례연구를 통해 제시된 아키텍처 컴포넌트들이 도메인의 핵심 요소이며 재사용이 용이한 구조를 이루고 있다는 사실도 확인하였다. 하지만 제시된 아키텍처 스타일이 일반화되어 모든 임베디드 시스템에 적용 가능하다는 사실을 확인하기 위해서는 많은 검증 작업이 뒤따라야 한다. 또한 임베디드 시스템의 영역 범위가 늘어가는 추세에 따라 현재 아키텍처 스타일도 더욱 세분화할 필요가 있다. 이와 병행하여 아키텍처를 비교 분석하여 성능을 평가할 수 있는 방법에 대한 연구도 수행해야 한다.

**[참고문헌]**

[1]최병욱,고경철,문전일,임계영 ,임베디드 리눅스, 홍릉과학출판사,P13~28,2002.2

[2]Stephen J Mellor, System Design: Architectures and Archetypes, Embedded System Conference, 2002.3.

[3]Morgon Bjorkander,Cris Kobryn, Architecting Systems with UML2.0, IEEE Software,P57, 2003.7