내장형 시스템에서의 객체지향 개발 프로세스 연구

오광근[0] 문전일 임계영
LG산전 중앙 연구소
{kkoh,jimoon,kylim}@lgis.com

A Study on the Object Oriented Software Developments Process in Embedded System

KwangKeun Oh[0] JeonIL Moon KyeYoung Lim
LG Industrial Systems R&D Center

요 약

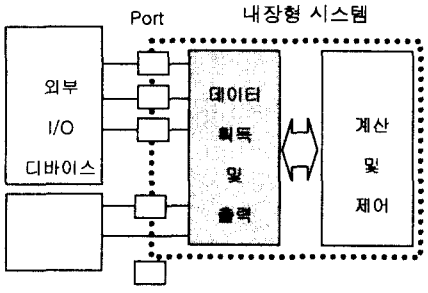
산업체에서도 다양한 고객의 요구 증가, 하드웨어의 발달 및 제품 life Cycle의 단축에 따라 내장형 시스템 개발에 재사용 컴포넌트를 이용한 효과적인 소프트웨어 개발 프로세스의 필요성이 대두되고 있다. 본 논문을 통해 내장형 시스템의 복잡성을 감소시키며, 재사용 가능한 부분을 컴포넌트화하여 관리할 수 있는 UML기반의 객체지향 소프트웨어 개발 프로세스를 제안한다.

1. 서 론

산업용 전기 전자 분야인 경우, 주로 하드웨어 중심으로 제품 개발이 진행되므로, 상대적으로 소프트웨어 부분에 대한 체계적인 개발이 이루어 지지 않았다. 하지만 고기능 사양 및 다기능 제품에 대한 요구가 날로 증가되므로써, 제품개발에 있어서 소프트웨어가 차지하는 비중은 점차 커지고 복잡해지고 있다.또한 소프트웨어에 대한 생산성과 품질보증에 대한 요구가 늘어나고 있는 추세에 있다[1]. 이런 요구들을 만족시키기위해 산업용 제품 분야에 대해서도 공학적인 접근방법에의한 소프트웨어 개발에 대한 방법론들이 다수 제안 되고 있다.하지만 실제 산업체에서 이런 방법론들의 적용은 매우 제안적으로 수행되고 있다[2]. 더구나 적용되는 방법론들 대부분이 절차중심의 구조적 방법론이 대부분이다. 이에 본 연구를 통해 내장형 시스템의 특성을 파악하여 효과적으로 컴포넌트를 추출 관리 할 수 있는 기준 아키텍쳐 중심의 객체지향 개발 프로세스를 제안한다. 특히 산업용 전기 전자 분야의 경우, 도메인 성격이 쉽게 변하지 않고, 시스템 고유 기능이 고정되있으므로 재사용 컴포넌트를 추출 관리하기 좋은 특징을 가지고 있다.

2. 내장형 시스템 분석

내장형 시스템은 대규모 시스템이나 제품군에 특정 기능을 제공하기위해 하드웨어와 소프트웨어의 조합으로 구축된 제품을 말한다. 산업용 전기 전자 제품으로는 PLC, Inverter등의 제품들이 해당된다. 대부분의 내장형 시스템들이 실시간성을 요하므로 실시간 시스템으로 다루어지기도 한다[3].일반적인 내장형 시스템은 그림1과같이 센서나 디바이스의 입력 신호를 받아 계산을 수행한 후 Actuator나 감시 디바이스들에게 제어신호 및 운전정보를 내보내는 구조로 되어있다.



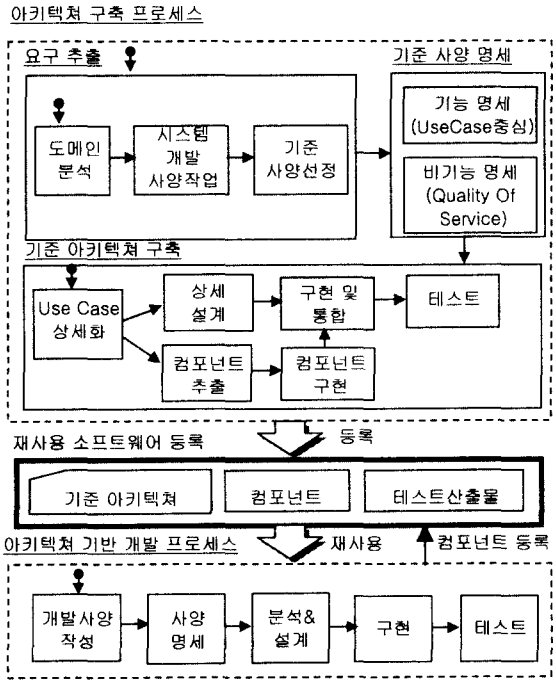
[그림1] 내장형 시스템 구조

내장형 시스템은 다른 컴퓨터 시스템에 비해 외부 디바이스들과 많은 통신을 통해 정보를 주고받으며, 시스템이 디바이스들의 정보에 의존하는 비중이 매우 크다. 이런 특징 때문에 개발시 외부 디바이스와 인터페이스하는 부분에 대한 모델링이 매우 중요한 위치를 차지하며, 독립적인 재사용 단위인 컴포넌트로 관리하였다.

3. 기준 아키텍쳐 중심의 개발 프로세스 모델링

내장형 시스템 개발 프로세스는 다음 3가지 기준에의해 프로세스 모델을 구축하였다.

- 모든 설계 요소들이 Use Case로부터 추적이 용이한 구조로 개발을 진행한다.
- 도메인의 재사용 가능한 요소들을 기준 아키텍처(Reference Architecture)를 구축한다.
- 외부 디바이스와의 인터페이스하는부분을 재사용 컴포넌트로 구축한다.



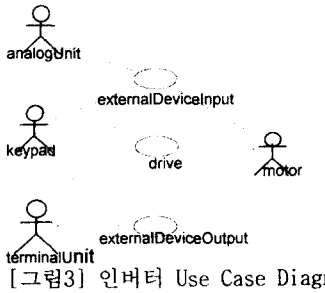
[그림2] 기준 아키텍쳐 구축 프로세스

그림2와 같이 개발 프로세스는 아키텍쳐 구축 프로세스와 아키텍쳐 기반 개발 프로세스로 나누어 진행된다.

기준 아키텍쳐(Reference Architecture)란 재사용을 전제로 구축한 소프트웨어 시스템을 말하며, 후에 재사용 기반 시스템 구축시 개발 기준요소로 작용한다. 기준 아키텍쳐 모델링은 4+1View모델을 기준으로 구축하였다[4]. 4+1 View란 시스템을 각 개발자의 역할에 따라 논리적관점, 프로세스관점, 구현관점, 배치관점으로 모델링하며 사용자관점인 Use Case모델로부터 앞의 4가지 관점의 모델링을 추적할 수 있게 아기텍쳐를 구축하는 방법이다. 먼저 요구 추출 단계에서는 도메인 분석을 통해 재사용이 가능한 도메인 영역을 추출하여 개발 기준 사양을 작성한다. 기준 사양 명세 단계에서는 Use Case 중심으로 기능사양을 명세화하고, 특히 내장형 시스템분야의 중요 개발 사양인 성능, 디자인 제약 조건과 같은 비기능 사양은 별도의 비기능 사양서를 통해 명세 작업을 수행한다. 기준사양 명세가 끝나면 Use Case명세서를 중심으로, Use Case 상세화과정을 거쳐 class를 추출한다. 이렇게 추출한 class들 중 독립적으로 수행 가능한 부분들은 별도의 설계 과정을 거쳐 재사용 컴포넌트로 등록한다.상세 설계 과정에서는 논리, 프로세스, 구현, 배치관점으로 아키텍쳐 모델링 작업을 수행한다. 상세설계후 구현 및 컴포넌트와의 통합작업,Use Case 시나리오 중심의 시스템 테스트를 거쳐 기준 아키텍쳐를 구축한다. 이렇게 구축된 기준 아키텍쳐는 동일 도메인 제품 개발 시 기준으로 사용된다.소프트웨어 재사용은 아키텍쳐 수준의 재사용과 컴포넌트 수준의 재사용의 두가지 수준에서 이루어 진다. 컴포넌트 수준의 재사용은 논리적 재사용 단위인 Subsystem을 통해 이루어진다. Subsytem은 재사용성이 고려된 고유한 특성을 가지고 있으며, 인터페이스 클래스를 통해 접근할 수 있다. 또한 Use Case로부터 추출한 시스템 수준의 테스트 케이스는 새로운 제품 개발시 재사용 할 수 있다.

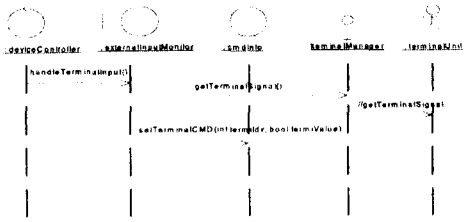
4. 사례 연구

인버터는 외부로부터 입력 주파수 지령 및 지령 신호를 받아 내부 계산을 통해 출력 주파수를 내보내는 대표적인 산업용 내장형 시스템이다. 본 연구에서는 인버터 시스템의 기준 아키텍쳐 모델링 및 재사용 컴포넌트를 추출 모델링하는 것을 기술한다.



[그림3] 인버터 Use Case Diagram

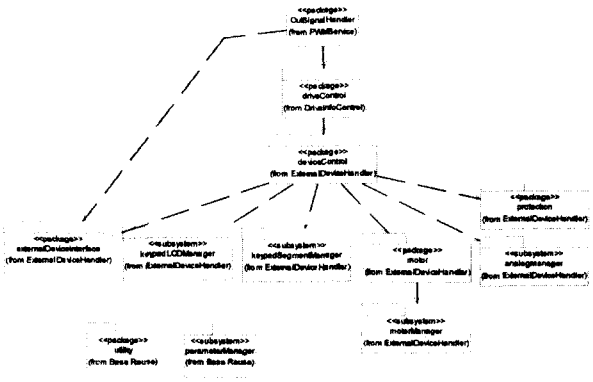
그림3은 기준 사양으로부터 기준 시스템의 주요 기능을 Use Case Diagram으로 나타 낸 것을 보여준다[5].



[그림4] Object간의 상호작용을 Sequence Diagram으로 묘사

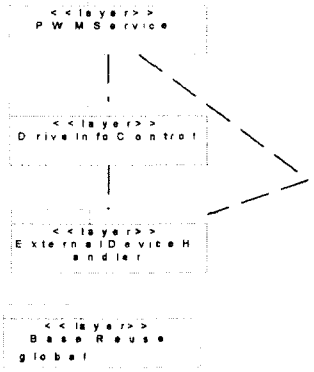
그림4와 같은 Use Case 상세화과정을 통해 Class의

인스턴스인 Object들과 그들간의 상호작용을 메시지 호출로 추출 한다.



[그림5] Package Dependency Diagram

이렇게 추출된 object들은 상세 설계 단계를 통해 그림5와같이 Package로 할당되거나 Subsystem으로 구현된다. 특히 외부 디바이스들과의 인터페이스로 모델링된 bject들 중심으로 subsystem을 추출 관리한다. 이렇게 추출된 Package들과 subsystem들은 그림 6와 같은 Layer 모델링을 통해 각 layer에 할당된다.

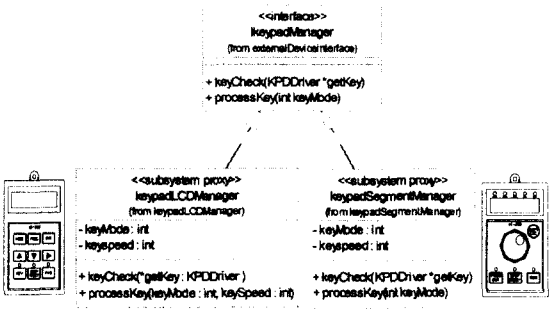


[그림6 인버터 layer 아키텍처

Layer	Class
External DeviceHandler	keypadLCDManager keypadSegmentManager motorManager analogmanager
Base Reuse	parameterManager

[표1] 재사용 하기위해 추출한 Subsystem들

각 Layer에 소속된 Subsystem들을 표1로 나타내었다. 외부 디바이스들인 키패드, 모터, 아나로그 입력단자의 데이터를 처리하는 부분들과 시스템 파라메터를 처리하는 부분들을 Subsystem으로 설계한 것을 나타낸다. 이렇게 구현된 Subsystem들은 인터페이스를 통해 서비스를 제공한다. 그림7은 LCD와 7Segment용의 두가지 타입의 디바이스를 가지고 있는 키패드에 동일한 인터페이스를 시스템에 제공하고, 필요시 선택적으로 사용 할 수 있게 모델링한 것을 나타낸다.



[그림7] 키패드 타입에 따른 Subsystem들

5.결론 및 향후 연구

본 연구를 통해 내장형 시스템 분야에 대한 객체지향 개발 프로세스를 구축하였다. 구축된 프로세스는 동종 제품 개발시 컴포넌트뿐만 아니라 기준 아키텍처 중심의 재사용을 제시한다. 또한 외부와 연결되어 각종 정보를 주고받는 디바이스 인터페이스들이 컴포넌트화 할 수 있는 좋은 특성을 가지고 있다는 것을 확인 하였다. 추후 연구가 필요한 사항으로는 기준 아키텍처 및 컴포넌트가 시스템 성능사양의 만족여부를 확인해야하며, 또한 기준 아키텍처를 이용한 제품 개발시 어느정도의 기간 단축 및 비용절감에 대한 효과가 있느지 검증해야 한다.

[참고 문헌]

[1] Lisa Brownsword,Paul Clements," A Case Study in Successful Product Line Development,CMU/SEI-96-TR-016,1996

[2] 전태웅, "실시간 소프트웨어 프로세스 정형화 연구", LG산전 위탁 연구결과 최종보고서, 1998

[3] 김문회, " Fault-Tolerance Systems and Real-Time Systems" ,LG산전 세미나자료, 1997

[4] Phillippe Kruchten," The Rational Unified Process An Introduction" ,Addison Wesley,2000

[5] Grady Booch,James Rumbaugh,Ivar Jacobson, " TheUnified Modeling Language User Guide" , Addison Wesley,2000