객체지향 기법을 이용한 전자연동 소프트웨어의 설계에 관한 연구

A Study on the Design of an Electronic Interlocking Software Using the Object-Oriented Method

이재호* 이병일* 박영수** 이재훈** 이기서***

Lee Jae Ho, Lee Byoung Il, Park Yong Su, Lee Jae Hoon, Lee Key Seo

ABSTRACT

Interlocking software used in the electronic interlocking has been implemented through the structured approach until now. But there was the demerit that interlocking software has some problem in the standardization and the maintenance because of the limit in structured approach. Object-Oriented method overcoming this demerit was specified in the design step and the analysis step. There were object model, dynamic model and functional model in the analysis step and there were also two steps which were system design and object design in the design step.

In this paper, Interlocking software was designed using Object-Oriented method to improve the standardization and the maintenance of the electronic interlocking. The electronic interlocking was analyzed with object modeling, dynamic modeling and functional modeling.

1. 서 론

연동장치는 신호기와 선로전환기를 제어하여, 열차의 안전하고도 효율적인 운행(출발·도착·입환 등)을 관리하는 장치이다. 연동장치는 열차의 고속화 및 고밀도 운전과 컴퓨터 기술의 발전으로, 계전기를 이용하던 계전연동장치에서 점차 컴퓨터 시스템을 이용한 전자연동장치로 대체되어 가고 있다. 현재까지 주로 사용되던 계전연동장치는 계전기를 신호기와 선로전환기 제어회로에 직접 삽입 사용하여, 그 동작으로 상호간에 필요한 연쇄 관계를 구현한 장치로 연동장치의 발전에 커다란 기여를 했다. 하지만 계전연동장치는 다음과 같은 문제점이 있었다. 먼저 각 역마다 선로의 배선의 모양이 서로 다르고 특수한 조건이 삽입되는 경우가 발생하면, 설계자의 개인적 차이에 따라 계전기 접점의 사용방법이 달라 표준화가 어려웠다. 또한 유지보수에 많은 시간과 노력이 요구되는 반면, 열차의 운행빈도와 속도가 높아져 유지보수 시간이 충분하게 주어지기가 어려워 유지보수에 누락과 오류를 발생시킬 수 있었다. 그리고 처리되지 않은 조작에 대해서는 응답이 없어 사용자가 무엇이 잘못인지 알 수 없으며, 전산화 등의 확장에도 어려움이 있었다. 전자연동장치는 다수의 계전기가 처리하던 내용을 프로그램으로 처리하는 것으로, 데이터 처리의 전산화가 가능하

* 광운대학교 제어계측공학과 박사과정

** 철도청 신호제어과

*** 광운대학교 제어계측공학과 교수

여 모든 처리 내용을 기록 및 자체 진단 할 수 있게 되며, 고장 발생 시 고장 메시지에 의해 고장 위치 또한 정확하게 확인 할 수 있었다. 그리고 고장모듈을 교체하여 신속한 유지보수가 가능하게 되었으며, 연동처리 프로그램에 각 역의 데이터를 처리하는 방식으로, 표준화 또한 가능하게 되었다. 하지만 이전에 연동소프트웨어는 구조적 접근방법 즉 구조적 프로그래밍과 자료구조에 중점을 둔 접근방식이 사용되어 다음과 같은 몇 가지의 문제가 있었다. 먼저 기능이 컴퓨터 상에서의 실현 단위인 것처럼, 분석 설계 되여 구현이 어려워, 최적의 접근 방법이 되지 못해 요구하는 사양과 구현된 시스템과 불일치가 발생하였다. 또한 자료에 대한 접근 제한이 전혀 없었기 때문에, 프로그래머가 자유스럽게 자료의 참조나 갱신을 수행할 수 있었으나, 대신 프로그래머는 자료에 접근할 때에는 그 구조를 모두 이해해야만 했고, 자료 구조의 변경 시에 따르는 처리 프로그램 변경의 파급범위를 제한할 방법이 없었다. 또한 코드의 일부를 변경하는 것 같은 확장에 대해서는 유사한 코드를 가진 복수의 절차나 자료 구조를 만들어야 했기 때문에, 기존 코드에 에러가 있으면, 복사·확장된 코드에도 똑같은 에러가 존재할 가능성이 높았다. 따라서 본 논문에서 이러한 미해결 된 문제들을 해결하기 위해 객체지향 접근 방법론을 이용하여 전자연동장치의 핵심인 연동소프트웨어를 분석·설계하여, 전자연동장치에 활용하고자 한다.

2. 본 문

2.1 객체지향방법론

객체지향접근은 객체(object)가 제각기 역할을 서로 분담함으로써 실세계와 같은 조직을 컴퓨터 상에서 재현할 수 있는 방법으로, 소프트웨어 개발방법이기에 앞서 실세계의 문제를 분석하고 소프트웨어의 체계를 잡는, 구현이 아닌 모델링(modeling)의 개념이다. 즉 객체지향 개발단계는 그림 1과 같이 객체지향 분석, 객체지향 설계, 객체지향 프로그래밍으로 나뉠 수 있다.

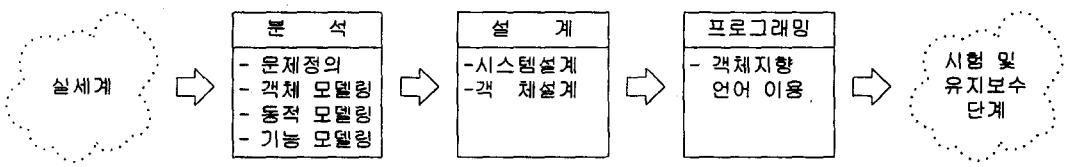

그림 1. 객체지향방법론의 개발단계
Fig 1. Development Step of Object-Oriented Methodology

객체지향 분석단계(OOA : object oriented analysis)는 문제에 관하여 객체를 규명하고 그들 사이의 관계를 찾는 단계로, 문제를 정의하고 실세계의 중요한 특성들을 객체 모델(object model), 동적 모델(dynamic model), 기능 모델(functional model)로 모델링 하여 실세계의 중요한 문제들을 보여주는 단계이다. 객체지향설계(OOD : object oriented design)는 문제를 해결하기 위한 접근단계로 성능의 최적화방법, 문제해결정책, 자원활용에 대해 결정하는 단계로, 이 단계에서는 시스템의 전체적인 구조에 대한 시스템 설계와 구현에 필요한 상세 설계인 객체설계로 나누어진다. 그리고 객체지향 프로그래밍(OOP : object oriented programming) 단계는 설계된 모형을 특정 프로그래밍 언어로 번역하는 단계로 객체·클래스·상속의 개념을 포용하는 객체지향언어가 가장 좋지만 일반적인 구조적 프로그래밍 언어로도 객체지향 개발에 활용될 수 있다.

3. 연동장치의 객체지향 분석(Object-Oriented Analysis)

연동장치를 아래와 같이 객체 모델, 동적 모델, 기능 모델로 구축하였다.

3.1 객체 모델의 구축

객체 모델은 객체를 그림으로 표현하여, 객체 클래스의 정적 구조를 나타낸 것이다. 객체 모델은 객체들을 식별하고 객체들간의 관계를 정의하고 각 객체 클래스의 속성과 연산기능을 보여주어 시스템의 정적구조를 표현하는 것이다.

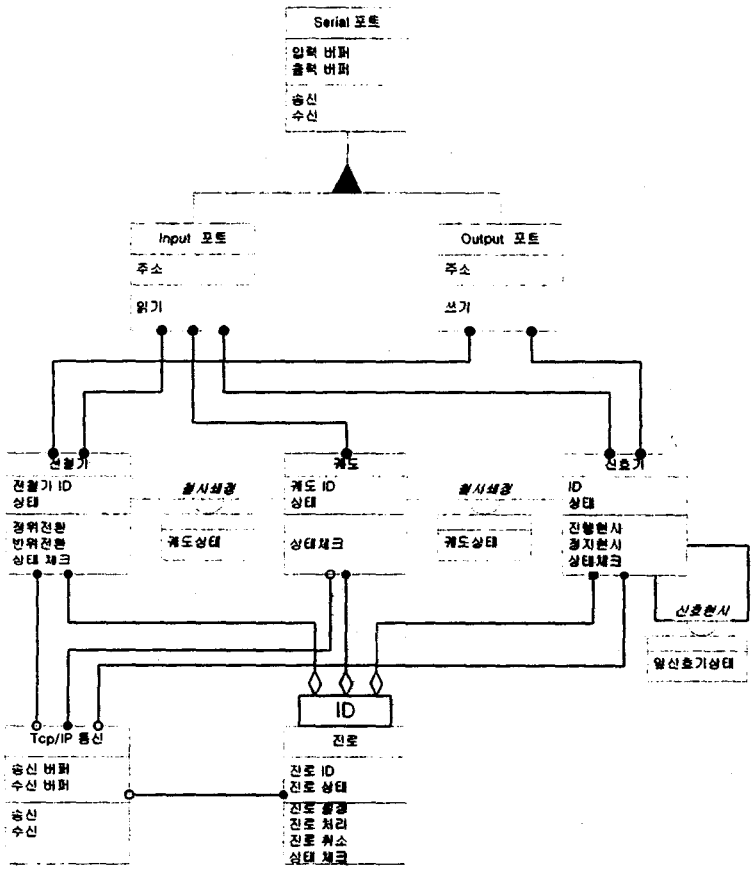

그림 2. 연동시스템의 객체모델

Fig 2. Object Model for Interlocking System

3.2 동적 모델의 구축

실세계의 물체는 상태를 가지로 있어, 시간의 흐름에 따라 상태가 변화한다. 따라서 실세계의 물체를 추상화한 객체도 상태를 가진다. 동적 모델은 시스템이 시간에 따른 행동과 시스템 내부에 있는 객체의 상태변화를 명확히 표현하는 것으로, 실시간 시스템의 경우에는 꼭 필요한 단계이며, 실시간 시스템이 아닌 경우는 이 단계가 불필요할 수도 있다. 아래 그림들은 진로설정에 대한 부분이다. 즉 그림 3는 진로설정에 대한 이벤트 추적도이고 그림4은 연동장치에서 진로설정에 대한 이벤트 흐름도이다. 그리고 그림 5은 이벤트와 상태간의 관계를 나타내는 연동장치의 상태흐름도이다.

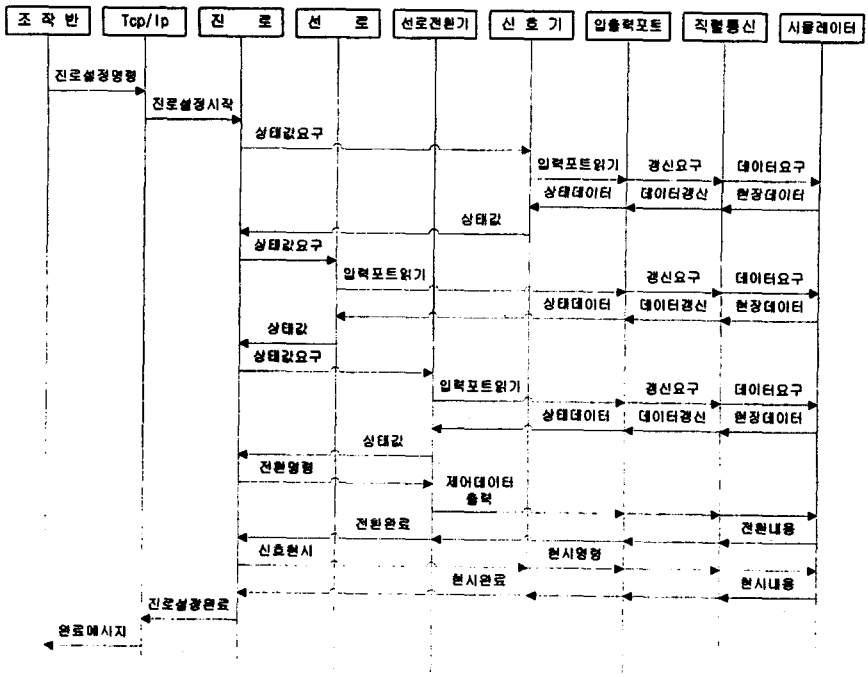

그림 3. 진로 설정의 이벤트 추적도

Fig 3. Event Trace for Route Set

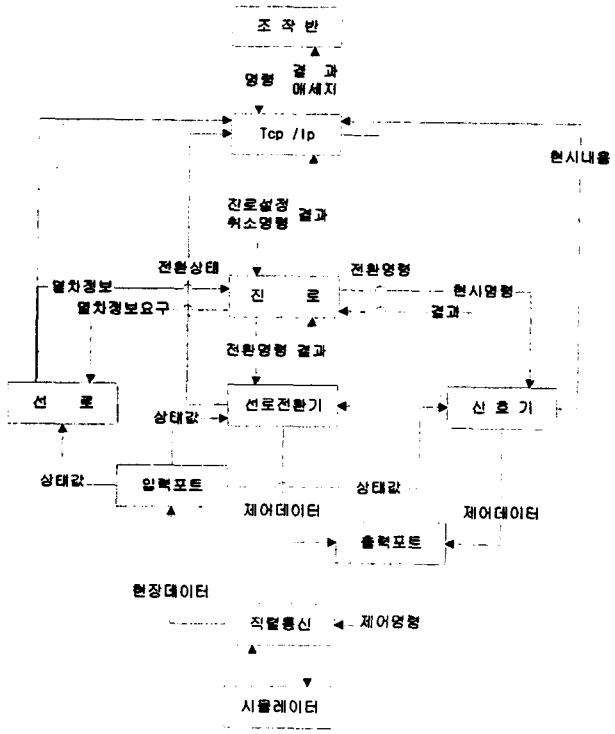

그림 4. 연동장치의 이벤트 흐름도

Fig 4. Event Flow diagram for Interlocking System

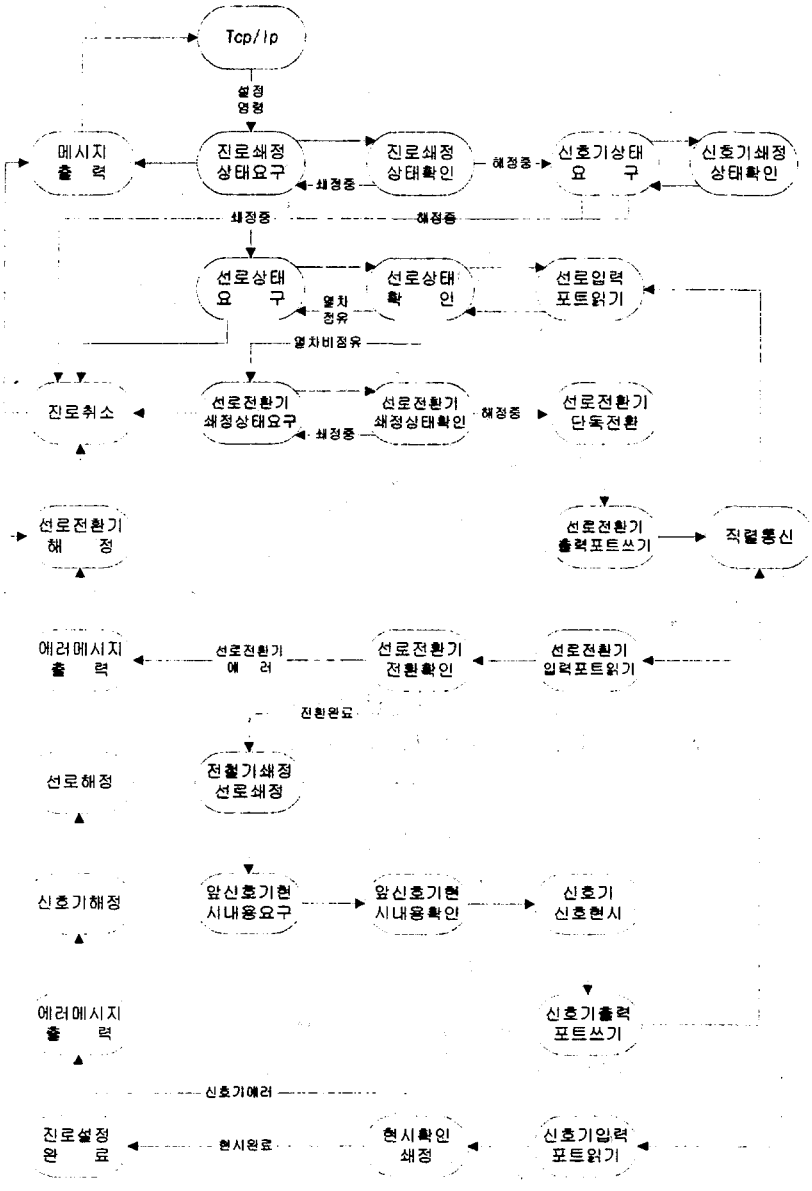

그림 5. 연동장치의 상태흐름도

Fig 5. State diagram for Interlocking System

3.3 기능 모델의 구축

기능 모델은 객체구조 등에 관계없이 값들이 어떻게 처리되는가를 나내는 것으로, 값이 조작되어지는 순서에 상관없이 조작되어진 출력 값이 입력 값으로부터 어떻게 유도되는가를 보여준다. 이 단계에서는 자료흐름도(DFD:Data Flow Diagram)가 만들어진다. 자료 흐름도는 구조적 분석에서 사용되는 자료 흐름도와 같은 것으로, 절차(process)와 자료흐름(data flow)과 정보저장소(data storage) 등으로 표현된다. 그림 7는 연동장치에 대한 자료흐름도이다.

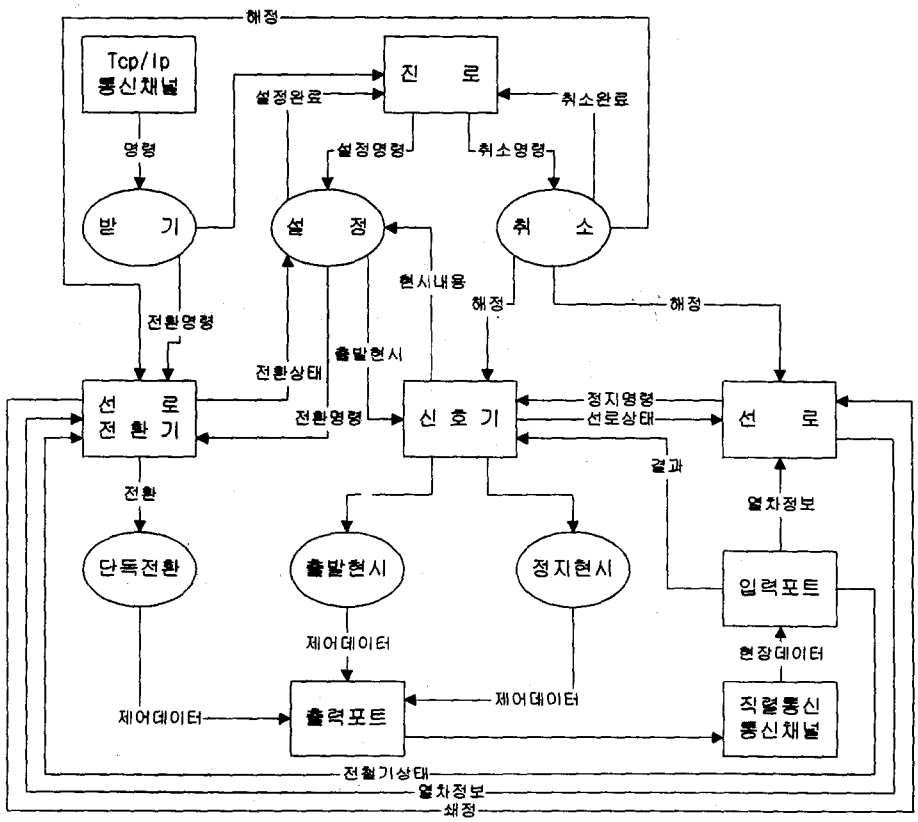

그림 6. 연동장치의 자료흐름도

Fig 6. Data Flow Diagrams for Interlocking System

4. 객체지향 설계(Object-Oriented Design)

객체지향 설계단계에는 분석단계의 출력물을 이용하여, 구현방법을 결정하고, 프로그래밍하는 데에 충분한 정보를 방법화하는 단계이다. 객체지향 설계단계는 기본설계인 시스템설계와, 상세설계인 객체설계 단계로 나눌수 있다.

4.1 시스템 설계

시스템 설계는 시스템의 대략적인 구조를 결정하는 단계이다. 이 단계에서는 객체 모델을 사용해서 시스템을 서브시스템으로 조직화(organization)하며, 하드웨어와 소프트웨어 컴포넌트에 서브시스템을 할당한다. 이 단계에서는 시스템의 하부구조 형성을 위해 동시성, 프로세서와 작업과의 관계 등의 중요한 의사결정을 하는 단계이다.

시스템 설계는 전체 시스템의 구조(architecture)를 결정하는 것으로 시스템을 기능, 물리적 위치, 하드웨어등을 고려하여 서브시스템으로 분리한다. 그림 7.은 연동장치의 시스템 설계도이다.

4.2 상세(객체)설계

이 단계에서 객체 모델의 구조는 효과적인 구현을 위해 최적화된다. 여기서는 각 클래스들의 기능수행을 위한 메소드들의 알고리즘 및 객체들간의 관계를 정확한 인터페이스로 정의하며, 구

현에 필요한 객체들을 추가시키고 자료구조를 성능 및 메모리 활용 측면에서 최적화한다. 즉 분석단계의 객체모형을 구체화하는 작업이 객체 설계이다.

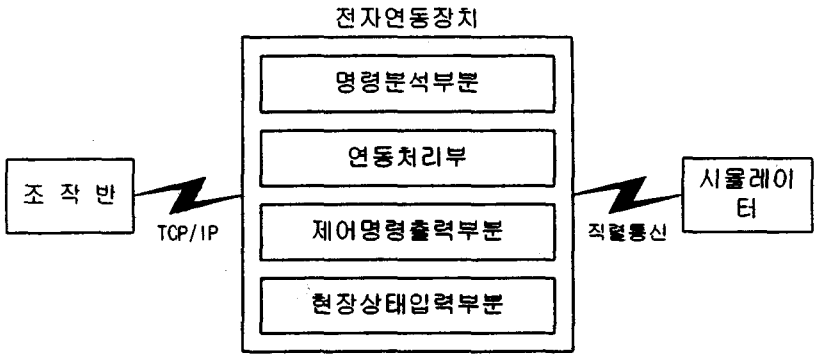

그림 7. 연동장치의 시스템 설계도

Fig 8. System Designed Diagram of Interlocking System

표 1. 설계된 객체

Table 1. Designed Object

선 로	신호기	전철기
- 현재상태 - 쇄정상태 - 소유 선로전환기 ID - 소유 신호기 ID	- 이전상태 - 현재상태 - 쇄정상태 - 해당궤도 - 앞 신호기	- 이전상태 - 현재상태 - 쇄정상태 - 해당 궤도 - 종류
- Get_현재상태 - Get_쇄정상태 - Set_현재상태 - 쇄정·해정	- Get_현재상태 - Get_쇄정상태 - Get_궤도상태 - Get_앞신호기상태 - Go·Stop - 쇄정·해정	- Get_현재상태 - Get_쇄정상태 - Get_궤도상태 - Set_현재상태 - 쇄정/해정 - 정위/반위전환

진 로	Tcp /IP	Port
- 진로상태 - 쇄정 배열 - 철사쇄정 리스트 - 진로쇄정 리스트 - 접근궤도 리스트	- 포트번호 - 주소의 길이 - 타임아웃 - 버퍼 - 사용된 소켓 개수	- 포트 1의 주소(PPA) - 포트 2의 주소(PPB) - 포트 3의 주소(PPC) - 콘트롤reg주소(CW)
- Get_현재상태 - Get_쇄정상태 - 진로설정 - 진로 처리 - 진로취소 - 쇄정/해정	- 소켓 열기 - 데이터 읽기 - 데이터 보내기 - 재접속 - 소켓 닫기	- 포트초기화 - PPA 읽기/쓰기 - PPB 읽기/쓰기 - PPC 읽기/쓰기

5. 결론

기존의 접근방법으로 개발된 소프트웨어의 문제점은 분석·설계단계의 미흡함을 극복하기 위해, 전자연동장치의 소프트웨어에 객체지향 방법론을 적용하여 분석·설계하였다. 그리고 선로, 선로전환기, 신호기처리 소프트웨어를 부품화하여, 역의 구성 요소 변경 또는 처리방법의 변경 등에 유연하게 대응하고, 소프트웨어의 재사용이 가능함을 알 수 있었다.

그리고 신호기, 선로, 선로전환기 객체들은 각각의 절차에 따른 처리 결과를 보증함과 전 단계

에서의 오류가 다음 단계로 확산되지 않음을 실험으로 확인할 수 있었지만, 연장장치에서 특수한 예외처리 부분에 대한 분석 및 설계가 미흡하여 다음과제로는 이러한 부분까지의 좀더 상세한 분석 설계가 이루져야 하겠다.

참고문헌

1. James Rumbaugh, Michael Blaha, William Premeplani, Frederick Eddy, William Lorensen "Object-Oriented Modeling and Design" ,Prentice-Hall, 1991.

2. I. Jacobson et al. "Object-Oriented Software Engineering : A Use CASE Driven Approach", Addison-Wesley, 1992.

3. Shaler S., and S. Mellor, "Object LifeCycle : Modeling the World in States", Prentice-Hall, 1992.

4. H. Yoshimura, S. Yoshikoshi, "Railway Signal", JASI, 1983.

5. "信號工學", 철도전문대학, 1989.

6. 윤정모, 한규정 역, "객체지향 시스템 개발", 동일출판사, 1996.

7. 이주헌, "실용 소프트웨어 공학론", 법영사, 1995.