

웹기반의 경량전철 유지보수 정보화시스템 개발방법론에 대한 연구

A Study on the methodology for Web-based Maintenance System Development of Light Rail Transit

이호용*, 한석윤**, 조홍식**, 조봉관**

Lee, ho-yong Han, seok yoon Cho, hong shik Cho, bong kwan

ABSTRACT

The methodology for maintenance system development should overall be considered with all of task characteristic, schedule, and the technical condition of investment sources. This research applies Waterfall model to methodology for the development and construction of maintenance system through customizing the CD (Custom Development) procedure of "HSDM (Hyundai System Development Methodology)" based on information engineering. The waterfall model called as classic life cycle paradigm is defined for the sequent development procedure, which make progress a next step after that a previous step is admitted with thorough review. This model which is popular and the oldest paradigm is used for system development by four steps; demands analysis, design, construction, and maintenance. This methodology advances higher abstract step from higher lower step using top-down approach from concept definition to construction, which notices the end of a step whenever a step is over. Therefore, each step is exactly divided, and consequently the output is clearly yielded.

1. 서 론

무인운전경량전철은 기계부품, 전기, 전자 및 제어, 정보통신, 재료, 토목 등의 종합적인 기술이 연계되어 있을 뿐만 아니라 차량시스템, 전력공급시스템, 신호 및 통신시스템, 선로구축물 등으로 이루어진 대형복합시스템으로서 무인운전을 전제로 한 시스템이므로 관련된 부품 및 시스템 개발에 있어서 특히 안전성과 신뢰성을 중요시하는 시스템 엔지니어링 기술의 적용이 우선적으로 요구된다.[1]

국내 최초로 구축된 고무차륜 AGT 시험선은 종합시험선으로 이를 이용하여 기술을 고도화 선진화하고 지자체에 대한 기술지원을 목적으로 건설하였다. 이러한 시설을 이용하여 경량전철 운영에 필요한 유지보수기준 자료를 확보하고 운영상에 필요한 유지보수 작업 효율 향상을 위해 유지보수 예방보전 및 전문가시스템 프로그램개발이 우선적으로 요구되었다. 이러한 시스템개발은 RCM(Reliability Centered Maintenance) 기반의 유지보수가 이루어지도록 지원하도록 시스템을 설계하여야 한다. 경량전철 유지보수 응용시스템의 특징은 유지보수응용시스템을 체계화하여 통합된 경량전철 유지보수응용시스템 업무 지원과 통합된 시스템 환경(서버 및 프레임워크) 구축으로 인한 도입기관의 시스템 유지보수 업무의 효율화 및 비용의 최소화하는 것이다. 유지보수 작업자의 인터페이스 및 일관된 Data표현을 통하여 사용자에게 시스템에 적응하는 시간을 단축하고, 다중의 업무를 시스템을 통하여 처리 시 통합된 시스템 이미지를 제공하도록 시스템개발을 추진하다. 또한 J2EE기반의 웹어플리케이션 아키텍처의 적용으로 재사용 가능한 유연한 시스템과 단위 시스템간의 유연한 인터페이스를 통한 통합 관리 가능하도록 설계하여 RCM 기반의 경량전철 유지보수가 이루어지도록 개발 방법론을 설계해야 할 것이다.[2][3]

* 한국철도기술연구원, 도시교통기술개발센터, 정희원

E-mail : hylee@krri.re.kr

TEL : (031)460-5715 FAX : (031)460-5749

** 한국철도기술연구원

2. 본 문

유지보수 정보화시스템 개발방법론은 개발의 업무성격과 일정, 투입자원들의 현황 및 기술력을 종합적으로 고려하여 적용되어야 한다. 또한 도시철도 유지보수 예방정비체계 구축방법론을 근거로 시스템을 개발한다.[4] 경량전철 유지보수응용시스템 개발 및 구축을 위한 방법론은 정보공학기반의 "HSDM(Hyundai System Development Methodology)" 의 CD(Custom Development)경로를 적용하여 폭포수형 모형으로 커스터마이징하여 적용한다. 고전적 라이프 사이클 패러다임이라고도 부르는 폭포수 모형은 일련의 개발 단계가 순차적으로 정의되어 있고 한 단계가 완결되어야 그 결과를 철저히 검토하여 승인과정을 거친 후에 다음단계를 진행하는 방식으로서, 요구사항 분석, 설계, 구현, 시험 및 유지보수의 순서로 시스템의 개발이 이어지며 가장 오래되고 널리 사용되는 패러다임입니다. 개념 정립에서 구현까지 하향식 접근 방법을 사용하여 높은 추상화 단계에서 낮은 추상화 단계로 옮겨가는 모델로서 각 단계가 끝날 때마다 과정의 끝을 알리고 그림 1과 같이 진행하기 때문에 단계별 정의가 분명하며 각 단계별로 결과물이 명확히 나오게 된다.[2]

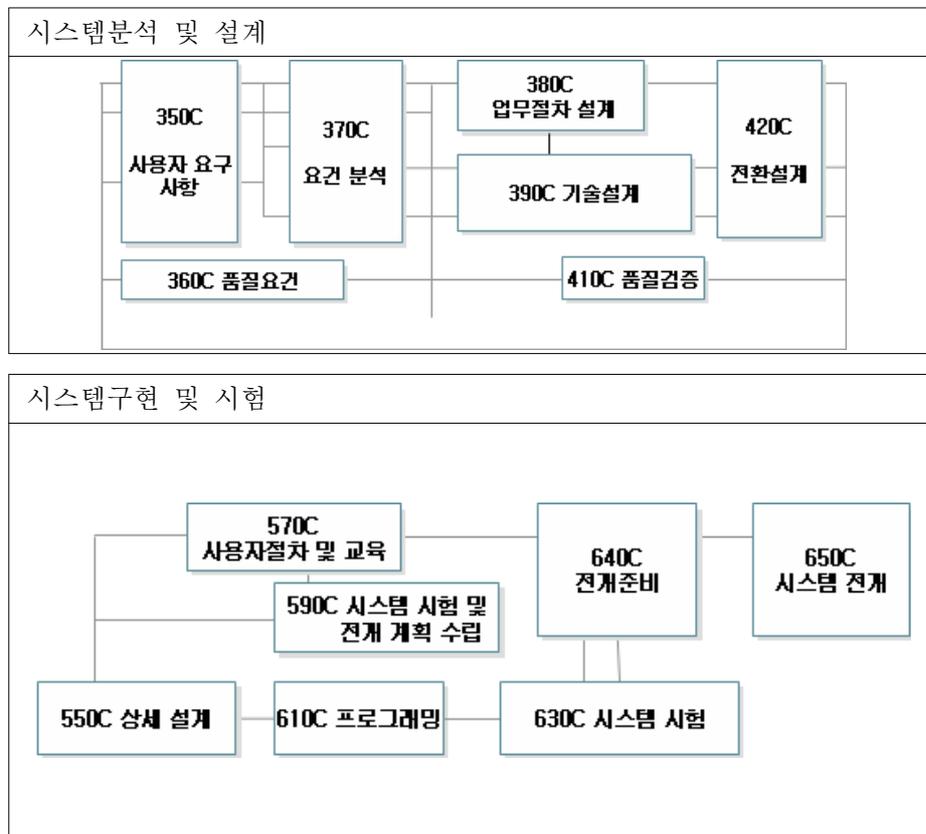


그림 1. HSDM Custom Development Path

2.1 단계별 프로세스 정의

단계별 프로세스는 개발방법론에 의해 계획단계, 분석단계, 구현단계, 인도단계로 구분되며 사업의 특성인 커스터마이징을 고려하여 구성된 전체적인 단계에 대한 구성 및 역할은 표 1과 같다.

표 1. 단계별 프로세스 정의

단계	태스크	작업내용
분석	요구사항 수집	요구사항 수집준비, 면담실시(수시) 요구사항 기술서 정리
	구축시스템 분석	컴포넌트 구조분석, 처리흐름 분석 User Interface 및 서버구조 분석

	요건분석	요구사항 분류, 기능별 요구사항 정리 요구사항별 구축시스템 비교
	시스템범위 선정	요건별 시스템 반영목록 및 통합범위 선정
설계	표준 및 절차수립	아키텍처산출물 및 개발표준수립
	통합대상 분석	통합대상 분석 및 목록 작성
	통합컴포넌트 설계	통합컴포넌트 모형작성 및 상세모형 작성
	통합 DB 설계	통합 DB 모형화 작성 및 ERD 정규화
	통합 UI 설계	화면구조설계 및 요건별 반영 목록 작성
	요구기능 설계	요구기능 설계, 기능별 반영목록 작성
구현	510C 통합 DB 구축	데이터베이스 공간 할당
	컴포넌트 구현	통합 플랫폼 연동 및 트랜잭션 명세 설계 및 검토
	통합 UI 구현	화면구축, 요구사항 UI 반영
	컴포넌트 테스트	컴포넌트 테스트 수행
	통합테스트	통합테스트 시나리오 작성 및 시나리오별 테스트 수행
	시스템 테스트	시스템 테스트 수행
인도	지침서 및 교육실시	사용자 지침서 작성 및 교육계획 작성, 교육실시
	인수테스트	인수테스트 계획 수립 및 실시

2.2 구현 표준 및 절차 지침

구현 표준 및 절차 지침에서는 스펙작성 지침과 동일한 구조를 가지고 Client/Server로 구분되어 Coding에 관련된 표준 및 Coding후 Test 절차까지를 지침으로 제시한다. 또한 구현표준에서 제시하는 경량전철 유지보수용시스템 개발 및 구축 프로젝트에 대한 Application에 대한 Architecture의 구현 처리 흐름은 그림 2와 같은 규약 속에서 Coding되도록 한다.

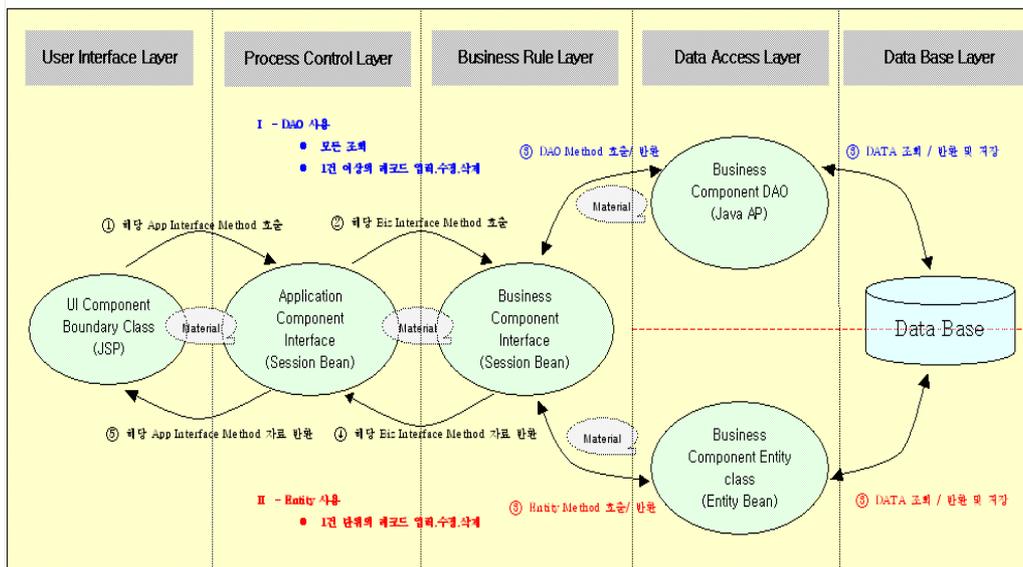


그림 2. Application 기술처리 흐름도

2.2.1 통합 표준

경량전철 유지보수응용시스템의 표준은 크게 User Interface 부문, Server부문, DB부문, 전자결재 부문, 도면관리 부문 및 Dynamic View기능 추가개발의 크게 6가지 부문에서 이루어 졌다.

각각의 부문별로 기존 고유의 기능이 필요한 부문에 대해서는 해당 기능을 유지하고 2개의 유사한 기능 및 구조들의 통합이 필요한 경우 1나의 기능 및 구조로 통합의 표준을 제시한다.

- 1) User Interface는 UI에 대하여 하나의 통일된 UI로의 통합제시를 통하여 공통된 사용 환경을 제시한다. 화면을 나타내는 타이틀 및 화면의 특정 기능을 수행하는 버튼을 상단에 표시하며, 화면을 구성하는 내역 및 기능버튼들을 그 하단에 구성하여 표시한다.
- 2) Server 통합표준은 디렉토리 구조 및 개발 FrameWork을 하나의 단일화된 구성으로 제공하기 위한 표준을 제시한다. 응용Application은 JSP서비스를 하기위한 응용 Application의 명칭을 krri_Apps라는 단일의 응용서비스 구조로 통합하며 HTML 및 이미지 서비스를 제공하는 Application의 명칭을 krri_htdocs라는 단일의 응용서비스 구조로 통합한다. 개발환경은 기존의 각각 구성되어 있던 개발을 위한 수행 Shell들을 다음의 명칭으로 통합하여 1나의 구조로 단일화한다. 서버개발환경 디렉토리의 구성을 dev_work의 명칭으로 단일화된 디렉토리로 통합한다.
- 3) DB 통합표준은 기존에 사용하던 고유의 TABLE명칭을 그대로 유지하고 하나의 통합된 계정으로 각각의 DB를 통합하고, 특정 TABLE들의 Data에 대하여 통합 혹은 인터페이스를 구성한다.
- 4) 전자결재 통합표준은 별도로 존재하는 결재목록 및 결재함, 진행함, 완료함 및 결재문서 조회를 통합하고 동일한 화면에서 일괄 처리가 가능하도록 결재구조를 통합한다.
 - 메인화면의 로그인 사용자 결재목록 통합
 - 결재함, 진행함, 완료함 결재목록의 통합
 - 다양한 조건에 의한 결재문서 조회 통합
 - 결재의뢰 및 결재처리 화면의 UI를 동일한 구조로 통합
- 5) 도면관리 통합표준은 BOM 관리 마스터를 구성하여 이를 바탕으로 자재 및 부품을 통합적으로 관리한다. 이 BOM을 바탕으로 각 서브시스템에서 동일한 코드를 바탕으로 사용한다.
- 6) Dynamic View통합 표준은 사용자 모니터의 해상도에 따라 화면의 사이즈를 동적으로 표시하여 화면의 가용성을 높인다. 가로방향에 대하여 화면의 사이즈를 Dynamic하게 표시한다. 또한 그리드(GAUCE 컴포넌트)가 표시된 화면의 경우 해당 그리드의 사이즈 Dynamic하게 표시한다.

2.2.2 Client 표준

Client 부분에 대한 표준은 크게 Client부분에서 구성되는 파일의 종류별(html, js, jsp)로 Coding하는데 있어서의 표준을 제시하고, User Interface인 화면을 Style별로 구분하여 해당 스타일에 맞는 Coding기법을 제시하고 있다. 또한, Client부분에서의 구현에는 UI Component인 Gauge를 적용하기 때문에 해당 컴포넌트를 여러 부분에서 적용하기 위한 기법들을 제시한다.

1) UI Architecture

가) Frame구조은 Client Source가 가지는 전체적인 구성을 정의하는 단계로 UI Architecture는 크게 3단의 Frame구조로 상단, 좌측, 우측으로 구분한다.

- 상단 frame : logo, 로그인 정보, 메뉴(1,2 레벨)
- 좌측 frame : 메뉴(3레벨 이상)
- 우측 frame : 업무 화면

나) 데이터의 흐름은 Client부분에서의 데이터 흐름을 정의하기 위하여 크게 화면에서 서버 호출과 서버 호출 결과를 화면에 표기 시로 구분되어 각각의 경위 대한 표준을 가진다.

다) 업무 흐름은 Client를 구성하는 소스인 html, js, jsp 간의 업무의 흐름을 가지기 위하여 전체적인 흐름을 표2와 같이 작성하도록 지침 한다.

표 2. 업무흐름과 소스구성 내역

업무흐름	소스 종류
- 사용자가 화면(html)상에 정보 입력 및 선택	(html)
- Request 전송을 위한 Event(마우스 클릭, 키보드 입력 등) 발생	(html)
- Event 별로 지정되어 있던 스크립트 실행	(html)
- 입력 정보에 대한 유효성 평가 및 사용자 권한 등을 체크	(js)
- 파라미터 값을 정비한 후 JSP 호출	(js)
- Request 값을 받아 분석하여 해당 로직 수행	(jsp)
- 서버단에 전송하기 위한 파라미터 setting	(jsp)
- 서버 메서드 call	(jsp)
- 서버로부터 받아온 데이터 중 화면단에 넘겨줄 것을 선별	(jsp)
- 스크립트 또는 Gauge 컴포넌트를 통해서 화면에 전송	(jsp, js)

2) 화면 파일 구성

먼저 제시한 Client부분에서의 파일 구조는 크게 3개의 파일(html, js, jsp)로 구성됨을 명시한다. 이에 각 파일별 명명규칙을 ‘프로젝트표준및절차지침서’에서 제시하였고, 이외에도 html Tag명칭, Component Object명칭, JavaScript Function의 명칭 등을 제시하여 동일한 Naming체계를 가지고 Coding할 수 있도록 한다.

3) 기능 구현 방법

기능 구현 방법에서는 실제적으로 html/js/jsp 상에 표기 방법을 구체적인 소스예제와 함께 제시하여, 개발자가 해당 예제를 참조로 Coding을 할 수 있도록 한다.

- Html상에서의 표기 방법
 - 프로젝트에서 적용할 html tag사용표기에 대한 규약사항 및 예시
 - Gauge Event Catch를 위한 Script 작성 방법 : Gauge 컴포넌트중 프로젝트에서 적용에 유용할 Component를 선별 하여 해당 컴포넌트를 적용하는 Script 제시
 - 또한 위의 Gauge Object를 작성하는 방법 제시
 - 위의 표기 방법을 적용한 화면구성 소스의 예를 주요 영역별 제시
- Js상에서의 표기 방법
 - 업무화면에서의 js 표기방법으로 스크립트 함수들과 그 설명을 담은 주석으로 구성
 - 공통모듈 운영 방법
 - 오류 메시지 처리 방법
- Jsp상에서의 표기 방법
 - jsp 파일의 구성
 - DataSet에서 데이터 읽는 방법
 - DataSet에 자료 작성 방법
 - Server Operation연동 방법 등 제시

2.2.3 Server 표준 및 절차

1) 명명규칙

명명규칙은 설계단계에서 표준을 수립하여 이를 적용하여 설계를 하였기 때문에 추가로 구현단계

에서는 구현을 위한 디렉토리 구조에 맞게 Package구조에 대한 명명규칙을 다시 정의하였을 뿐이다.

2) 프로그램 작성규칙

구현 단계에서 사용하는 code의 작성 규칙을 정의 하고 소스코드의 일관성, 빈번한 공통적인 에러 발생 감소 및 가독성 및 유지보수 편리성 등의 목적을 가지고 제시한다. 썬마이크로시스템즈의 자바 언어 명세(Java Language Specification)에서 소개하고 있는 자바 언어 코딩 표준(Java language coding standards)을 참조하여 표준을 제시한다.

- 가) Java 프로그램 파일 구성은 각 구성 요소들은 줄띄우기를 통해 구분이 용이하도록 프로그램 주석문, package 선언문, import 문, 클래스 선언 등으로 구성한다.
- 나) 주석문 규칙(Comment Guide)은 프로그램에서 필요한 부분에 설명을 하고자 할 때, 주석문을 주게 되는데 컴파일시는 무시된다. 이에 대한 주석문 처리 기법을 ‘클래스 주석문’, ‘블럭 주석문’, ‘행 주석문’으로 구분하여 작성한다.
- 다) 프로그램 Log작성 규칙은 프로그램의 시작과 끝에 Web Application Server 및 개발환경에 따라 Log를 기록하는 코드를 삽입한다.
- 라) 선언 규칙은 자료선언 순서를 표준화 한다. 또한 short, int, long, float, double, char, byte, boolean 등의 순서로 사용하고 클래스 타입 선언은 기본 자료형 선언 이후에 사용한다. 그러나 프로그램의 중간에서 변수 선언을 하여 사용하지 않도록 한다.
- 마) 띄우기에서는 ‘줄띄우기’, ‘탭 띄우기’, ‘여백 띄우기’등 올바른 표기법과 잘못된 표기법을 예시로 설명한다.
- 바) 제어 규칙은 Java 에서 제공하는 제어구조에 대한 작성 표준을 기준으로 하고 선택구조(if)와 반복구조(do while 형식, switch형식, go to의 사용, try catch finally)를 사용한다.

3) 시스템 개발 환경

개발환경부분에서는 개발자용 PC Directory구조와 Server의 디렉토리 구조를 정의하여 개발후 서버에 해당 데이터를 ftp등으로 등록하고, 테스트하는 과정을 가진다. 특히, 서버의 Directory구조는 개발팀에서 정의한 기반구조를 적용한 디렉토리 구조를 가지도록 한다.

4) 개발 절차

개발절차는 실질적으로 서버단의 코딩을 시작해서 테스트까지 적용해보는 흐름도를 제시하여 개발자에게 이해를 도모한다.

- 가) Entity Bean 생성은 EntityBuilder 프로그램을 실행하여 설계한 DB구조에서 원하는 Entity를 선택하여 EntityBean을 생성시켜 Coding할 수 있도록 한다.
- 나) 코드작성은 EntityBuilder에 의해 생성된 소스중에 Bean폴더안에 xml파일 생성이 필요하다. 해당 xml파일은 Component단위로 하나의 파일을 생성하고 해당 컴포넌트에 속하는 모든 Interface에 대하여 정의하도록 샘플소스와 함께 설명한다.
- 다) 서버에 업로드된 개발자PC에서 생성한 소스를 서버에 등록하기 위한 절차를 수립하고, 테스트 할 수 있도록 개발자ID별 디렉토리를 생성시켜주고 Pwd를 개별로 관리하도록 한다.
- 라) EJB Gen은 서버에 Upload시킨 소스를 테스트하기 위하여 공통적으로 생성한 Builder 프로그램에 의해 소스를 Build시킨다. 또한 Build의 순서는 Builder 프로그램에 의해 자동 순서적으로 실행을 처리하지만, 개발자에 의해 제어되는 순서는 우선적으로 'DAO' ->'EntityBean' ->'Interface' 순으로 처리할 것을 명시한다.
- 마) EJB Deploy은 Builder시킨 Application을 Install시키기 위한 Lib구조를 사용한다. 공통 라이브러리 shell 프로그램에 의해 다음과 같은 처리를 할 수 있다.
 - appinst.sh : 해당 하는 공유 라이브러리를 Mapping
 - appstat.sh : Deploy상태 점검
 - appstart.sh : application 실행 시작
 - appstat.sh : 정상적으로 start되었는지 확인

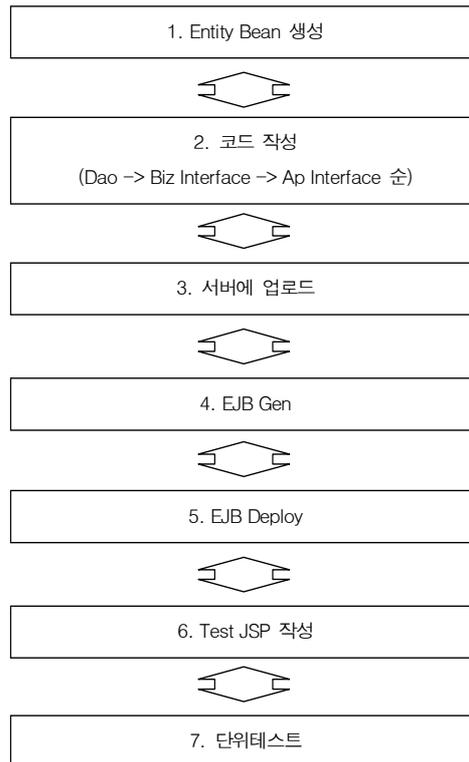


그림 3. 개발순서 정의

2.2.4 DB 운영 및 SQL문 작성 표준

설계단계에서 제시한 DB 모델링 표준을 근거로 작성된 ERD(Entity- Relational Diagram)으로 부터 DBA에 의해 DB Creation을 시킨 후 운영을 하도록 한다. DBA는 ‘DB운영규칙’을 작성하여 각 개발자에게 교육훈련을 실시한 후, 개발별 권한 설정과 DB운영기준을 제시한다. 또한 개발자에 의해 소스 상에 작성되는 SQL문을 생성시 일정한 규약과 규칙을 적용하여 일관성 있게 관독할 수 있어 유지보수에 용이하도록 한다. 특히, 설계단계에서 제시한 ‘DB설계 명명규칙’에 의해 생성된 Entity들을 모두 물리적구조로 변경하여 Table화 했으며, 논리적설계와 물리적설계 일부로 설계된 설계단계 산출물을 근거로 구현초기에 물리적 DB로 구축을 완료한다. 업무별 담당자들에 의해 작성된 DAO Spec에 의해 각 Table별 Index등을 생성하고, 각 테이블스페이스에 대한 규약을 가진다.

1) DB 운영 규칙

DB 운영 규칙에서는 프로젝트에서 선정한 DBMS인 Oracle 9.0.2를 설치에서 부터 백업의 순서로 관리방안을 작성하고 전체적인 구성은 다음과 같다.

- 설치전 준비사항 : 시스템 사용현황 확인, 업무서버 필요 서버 파일시스템 확인
- 파일 시스템 구성
- Oracle 설치 : Oracle을 설치할 파일시스템 확인 -> Oracle 시스템 user의 생성 -> Oracle User의 .profile설정 -> Listener.ora파일 수정
- DataBase 구축 : 테이블스페이스 생성 -> SysUser관리 -> Oracle 사용자 생성 -> 테이블 (Table)생성 -> 뷰(View) 생성 -> 권한(Grant)부여. 시노임(Synonym)생성 -> 외래키 (Foreign Key) 생성 -> 인덱스(Index) 생성 -> 오브젝트(Object) 생성 -> 초기데이터 구축
- DB 관리정책 : 오라클에서 제공하는 System 진단 tool, Oracle Tuning
- 백업정책 : 백업 윈도우의 설정, ARchiving Redo Information, BackupW Methods, Recovery, Case Study

2) SQL 작성 규칙

가) 코딩규칙(SQL)은 SQL문장의 Syntax 및 작성법의 표준을 준수함으로써 읽기와 편집의 용이함과 문장의 동일함을 추구하여 DB Level에서 재 Parsing이 발생하지 않도록 하고, 프로그램 성능을 저해하는 쿼리기능을 사전에 방지함으로써 프로그램의 성능을 향상시켜 차후 쿼리튜닝 소요시간을 단축할 수 있도록 하기 위해 SQL코딩규칙을 제시한다. 또한 원활한 유지보수를 위해 복잡한 쿼리는 지양하도록 한다.

- SQL문에 대한 기본적인 Coding Rule과 함께 Alias관리, Insert문, Update문, Delete문들에 대한 예제를 통하여 잘못된 구분을 지적하고 효율적인 SQL을 제시한다.
- 인덱스 컬럼 선정 기준 및 고려사항들 제시
- SQL문에 대한 기본적인 체크사항들
 - > Select절, From절, Where절, Order By절, Group By절, Having절별로 자체해야할 문장들을 제시한다.

나) 활용예제는 다음의 각 주제별로 활용예제의 SQL을 제시하여 Coding에 참조할 수 있도록 한다.

- Sort를 대신하는 Index 예제, Max처리 예제
- Exists처리 예제 및 1:M Join의 부분범위 유도
- Index Column의 변형
- Not Operator : 부정비교는 가급적 사용을 금하고 Not Exists등을 사용하여 긍정적인 비교를 사용
- Null, Not Null : Not Null비교는 「> 0」, 「> 」로 비교

2.3 공통 API

시스템 개발시 개발자들이 공통적으로 사용되는 모든 API들을 정리하고 공통 API는 이후에도 계속적으로 추가 생성되어 개발자들이 통합되고 일관성있는 Method들을 적용하여 편리하고 표준화된 코딩이 될 수 있도록 한다. 공통 API에는 Clinet 단과 Server단의 파일구조별로 사용할 수 있는 Method로 정리한다.

3. 결 론

무선통신을 기반으로한 고무차륜형 무인운전시스템을 개발하여 경산시에 소재한 시험선을 활용하여 신뢰성 시험평가를 실시함으로써 세계적인 수준을 확보하는 것이다. 이러한 목표를 이루기 위해서는 실제 경량전철 운영에 필요한 유지보수 기준 자료 확보 필요하며, 이러한 자료가 확보될 수 있는 시스템 마련이 무엇보다 중요하다고 할 수 있고 향후 경량전철을 도입하고자하는 지자체나 운영기관에게 좋은 연구결과를 제시할 수 있을 것으로 본다. 체계적인 개발방법론을 적용하여 무인운전경량전철 유지보수 예방보전 및 전문가시스템 프로그램개발로 운영상에 필요한 유지보수 작업 효율 향상을 이루어 RAMS(Reliability, Availability, Maintainability and Safety) 향상될 것으로 기대된다.

참고문헌

1. 이안호(2005년), “경량전철시스템 기술개발사업 7차년도 연구결과보고서(종합시스템엔지니어링)”, 한국철도기술연구원.
2. 이안호(2006), “무인경량전철시스템 RAMS향상 연구”, 한국철도기술연구원.
3. 박기준(2005), “도시철도유지보수체계정보화연구”, 한국철도기술연구원.
4. 이호용, 한석윤, 박기준, 배철호, 서명원(2004년), “도시철도유지보수 예방정비체계 구축방법론에 관한 연구”, 논문집, 제7권 3호, pp245-250, 한국철도학회