GOVP1200940976

미래철도기술개발사업 제5차년도 최종보고서

R&D/03교통핵심A02

대형·고밀도운행 철도역 자동운행 제어시스템 개발 연구보고서

2008. 12. 31.

주관연구기관 / 한국철도기술연구원
협동연구기관 / 대아티아이(주)

국 토 해 양 부
한국건설교통기술평가원

Construction & Transportation R&D Report

제 출 문

국토해양부장관(한국건설교통기술평가원장) 귀하

본 보고서를 "대형·고밀도운행 철도역 자동운행제어시스템 개발"의 최종보고서로 제출합니다.

2008. 12. 31.

주관연구기관 : 한국철도기술연구원

연구원장 최 성 규

협동연구기관 : 대아티아이(주)

사장 최진우

■ 총 괄 연 구 책 임 자 / 책임연구원 홍순흠 연구총괄

■ 주관연구기관 참여연구원 / 선임연구원 김영훈 외부테스터설계

/ 주임연구원 김경민 알고리즘개발

■ 협동연구기관 참여연구원 / 선임연구원 이영수 협동연구총괄

/ 연 구 원 박성규, 안진, 고재호 테스트시나리오작성

/ 연 구 원 용성완, 김민수, 장일수 핵심서브시스템제작

/ 연 구 원 하승태, 김덕환, 조우식 서브시스템개발

/ 연 구 원 김우정, 박범수 외부테스터제작

/ 연 구 원 김성주, 강경모, 유진성 종합시험센터개발

/ 연 구 원 한찬희, 이안준, 박정호 종합시험센터개발

요 약 문

I. 제목

○ 대형·고밀도운행 철도역 자동운행제어시스템 개발

II. 연구목적

○ 본 연구개발사업의 최종목표는 철도역단위 열차운행 제어시스템의 개발이다.

○ 이는 현재 중앙 집중제어 방식에 의해 운영되는 역에서의 수작업 열차경로제어 업무 자동화, 센터 운영모드와 역 운영모드에 따라 발생하는 문제점을 해결하기 위해 역 분산제어 및 운행협조가 가능한 시스템 구축, 시스템의 변경·추가·유지보수의 어려움을 해결하기 위해 온라인 확장이 가능한 역시스템의 구축, 열차지연 및 외란상황의 효율적 대처가 가능한 시스템의 구축 개발을 포함한다.

○ 당해연도 연구목표는 핵심서브시스템 제작 및 외부테스터 개발과 개발된 역시스템을 테스트 할 수 있는 종합연계시험센터 개발 및 시험이 있으며 시스템 테스트를 위한 현장적용시험이다.

III. 연구개발의 내용 및 범위

구분	연구개발 목표	연구개발내용 및 범위
5차년도 (2007)	대형역 자동운행 제어 서브시스템 개발 및 시험	○ 핵심서브시스템 제작 및 시험 · 왕십리, 성북역 핵심서브시스템 제작 · 홈 리스케줄링 Upgrade 및 ASCS 통합 · 역간 및 센터연동시험 ○ 서브시스템 개발 및 시험 · 라인모니터링 서브시스템 개발 · 수송관리 서브시스템 제작 · 라인경합 서브시스템 개발 ○ 외부테스터 개발 · 외부테스터 상세설계 · 외부테스터 제작 · 외부테스터 자체 기능 시험 · 외부테스터 온라인 테스트모드 시험 ○ 홈 리스케줄링 연동시험

		· 홈 리스케줄링 자체 기능 시험 · 인접역 연동시험 · 라인경합관리 연동시험(센터연동시험) ○ 종합연계시험센터 개발 및 시험 · 종합연계시험 계획 작성 · 현장 실 데이터 DB 구축 · CTC 에뮬레이터 서버와 타시스템 I/F 개발 · CTC 에뮬레이터서버 시험 및 기능보완 · 에뮬레이터 보완 제작 · CTC L/S 모니터링 콘솔 개발 · 종합연계시험센터 설치 · 종합연계 모의 시험 · 결과분석 및 수정보완 ○ 현장시범시스템 구축 및 시험 · 현장시범시스템 구축 · 현장시험 · 시험결과분석 및 기능보완

IV. 연구개발 결과

1. 국·내외 현황조사 및 시스템 요구사항 분석

○ 국내 운행제어시스템의 사례조사는 열차집중제어장치를 조사하였다. 통합 CTC(Central Traffic Control)시스템은 각 지역사무소에 분산 배치되어 있는 사령 설비를 한 곳으로 통합하여 주요 선로구역의 열차 운행 관리와 열차 운행 시각, 유지보수 등을 집중관리 한다.

○ 국외현황 분석은 일본 도쿄 도심지역의 자율분산 열차운행 제어시스템인 ATOS(Autonomous decentralized Transport Operation control System)에 대한 기술개발 현황 중심으로 조사하였다.

○ 국·내외 운행제어시스템 현황 분석은 국내 시스템은 집중화에 대한 시스템이며 국외 시스템은 집중화 시스템과 분산시스템을 함께 사용하고 있으며 세부기술적인 측면에서는 국내 기술은 동등한 기술을 보유하고 있지만 자율분산 시스템 기술을 적용한 국외 사례에서의 온라인 확장 및 온라인 유지보수 기능들은 도입하면 시스템 구축비용이나 유지보수 비용 절감 효과를 기대할

수 있다.

- 청량리역, 망우역 등의 현장조사를 통해 사용자의 요구사항을 열차운용 담당자와 신호취급 담당자, 수송취급 담당자의 요구사항을 도출하였다.
- 연구개발에 도입할 자율분산 시스템 기술의 요구사항은 온라인 확장, 역 자율성 확보, 자율제어 능력, 상호협조 운영, 온라인 유지보수 등의 요구사항을 도출하여 시스템 개발에 반영하였다.

2. 대형·고밀도운행 철도역 자동운행제어 시스템

가. 대형 · 고밀도운행 철도역 자동운행제어 시스템 구성

- 대형 · 고밀도운행 철도역 자동운행제어 시스템(Autonomous Decentralized Traffic Management System, ADTMS)의 구성은 아래 그림 1과 같이 전체 시스템을 자율분산 열차운행관리시스템으로 명명하고 역에 설치되는 시스템은 자율분산 역제어시스템, 센터에 설치되는 시스템은 자율분산 노선관리시스템으로 하였다.
- 자율분산 노선관리시스템은 라인모니터링 서브시스템, 라인경합관리 서브시스템, 라인정보 서브시스템으로 구성되며 자율분산 역제어시스템은 실시간 열차추적 서브시스템과 경로제어 서브시스템, 열차모니터링 서브시스템인 3개의 핵심 서브시스템으로 구성되고 지원서브시스템 수송관리 서브시스템으로 구성된다.

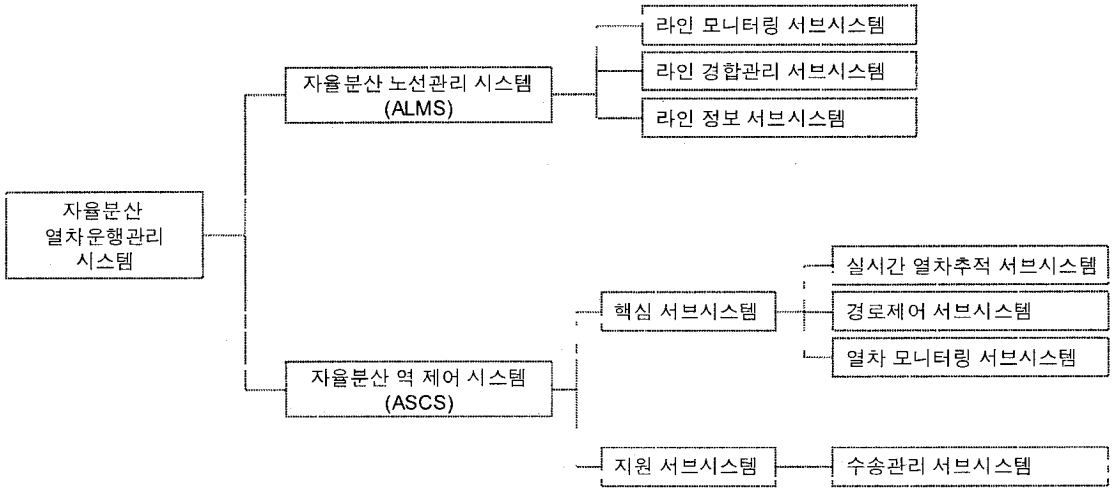


그림 1 자율분산 열차운행관리 시스템 구성도

○ 자율분산 열차운행관리 하드웨어 구성도는 아래 그림 2와 같다. 라인단위의 라인센터에는 3개의 서브시스템이 있고 역의 각각의 서브시스템에는 자율분산관리자와 각각의 기능을 수행하는 프로세스가 존재한다.

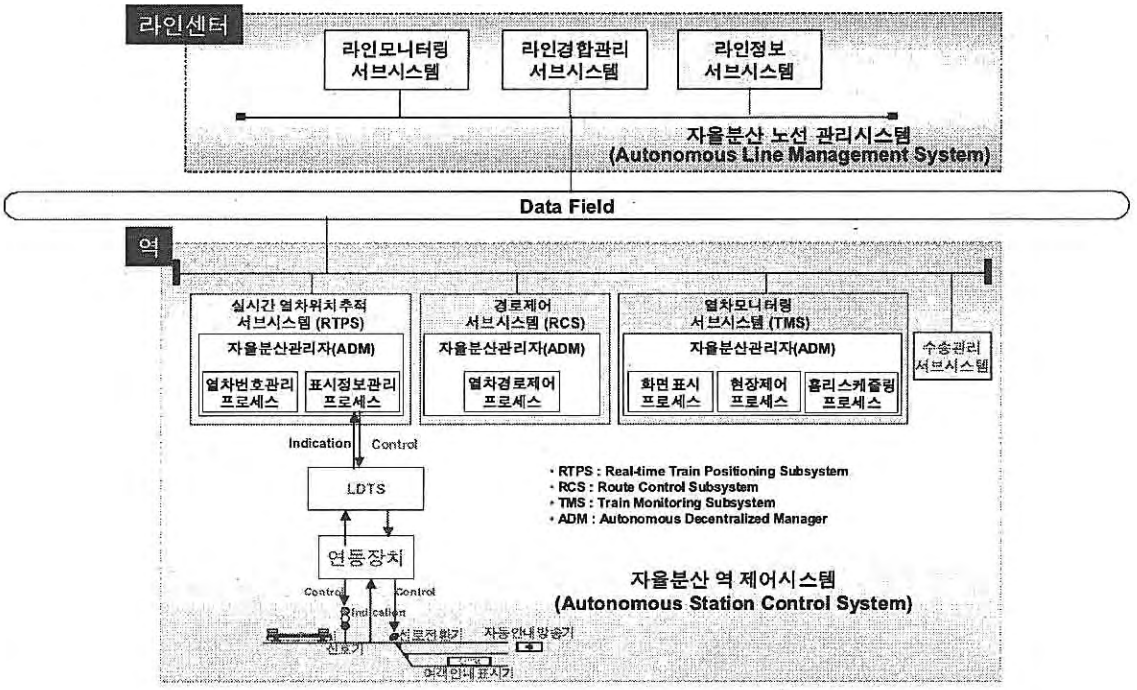


그림 2 자율분산 열차운행관리 하드웨어 구성도

나. 자율분산 역제어시스템

자율분산 역제어시스템(Autonomous Station Control System, 이하 ASCS)은 세 개의 핵심서브시스템과 수송관리 서브시스템으로 구성된다.

(1) 실시간 열차위치추적 서브시스템

○ 핵심서브시스템의 하나인 실시간 열차위치추적 서브시스템(Real-time Train Positioning Subsystem, RTPS)은 역내에서 운행하는 모든 열차들에 대한 열차의 위치 추적과 열차번호를 관리하기 위한 기능을 가지며, 역정보전송장치(LDTS; Local Data Transmission System)로부터 수신되는 선로변 장치들(신호기, 선로전환기, 궤도회로 등)의 상태를 기록한다.

○ 실시간 열차위치추적 서브시스템은 기능의 특성에 따라 열차번호관리 프로세스, 표시정보관리 프로세스로 구성된다.

(2) 경로제어 서브시스템

o 경로제어 서브시스템(RCS; Route Control Subsystem)은 실시간 열차위치추적 서브시스템의 표시정보관리 프로세스와 열차번호관리 프로세스로부터 현장 선로변 시설물의 논리적 상태정보와 열차의 이동정보(위치, 열차번호, 방향 등)를 수신한다. 수신된 정보를 이용하여 자율분산역내의 특정한 현장조건(지정된 궤도회로의 점유 등)이 만족하고 제어모드가 역 자동 제어모드인 경우, 열차스케줄로부터 해당 열차의 진입정보를 통하여 현장의 연동조건과 시설물의 상태를 판단하여 진로제어 명령을 자동으로 생성하게 된다.

o 자율분산역의 본선에 진입한 열차 중에서 열차조성 등의 입환 작업이 필요한 경우, 열차를 입환 스케줄과 비교하여 입환선 또는 대피선으로 자동진로 설정을 제어하게 된다. 설정되는 모든 진로제어 정보는 LDTS로 전송되며, LDTS로부터 해당 정보를 수신한 연동장치에 의해 진로가 설정된다.

(3) 열차모니터링 서브시스템

o 열차모니터링 서브시스템(TMS : Train Monitoring Subsystem)은 자율분산 역제어시스템에서 운영자를 위한 MMI(Man Machine Interface)를 지원하며, 실시간 열차위치추적 서브시스템으로부터 수신되는 선로변 시설물의 표시상태, 열차번호 이동 상태 등을 이용하여 화면상에 표출하기 위한 기능을 가진다. 선로변 시설물에 대한 장애 상태를 알리기 위한 경보 기능, 이상 발생 시의 경고 기능 등이 운영자에게 제공되며, 필요시 운영자는 자율분산 역에 대한 수동 제어를 실행할 수 있다.

o 열차모니터링 서브시스템은 기능의 특성에 따라 화면표시 프로세스와 현장제어 프로세스, 홈 리스케줄링 프로세스 (Home Rescheduling Process, 이하 HR) 등으로 구성된다.

(4) 수송관리 서브시스템

o 수송관리 서브시스템(SMS; Shunting Management System)은 현장역에 설치되어 운영되는 ASCS를 지원하는 서브시스템으로써 역의 수송원이 취급하는 입환 관리 등의 역 단위 열차운행과 관련된 업무를 시스템화하는 설비이다.

o 수송관리 시스템의 주목적은 역 단위의 열차 운행계획에 따라 역 구내에서 운

용되는 입환 스케줄을 생성 및 변경하여 입환 지시서를 발행하고 타 서브시스템이 입환 스케줄이 결합된 온라인 스케줄을 사용하게 한다. 또한 현재 운행중인 온라인 스케줄의 조회 및 본선스케줄 조회 기능도 수행한다.

(5) 자율분산관리자

- 자율분산관리자(ADM; Autonomous Decentralization Manager)는 하나의 서브시스템이 타 서브시스템의 영향 없이 자율성을 확보하여 자율제어를 할 수 있도록 하는 기능과 데이터필드를 통해 서브시스템간의 자율협조를 할 수 있도록 하는 주요 기능으로 구성되어 있다.
- 자율분산관리자의 주요 기능은 DF(Data Field) 수신 및 데이터 구동, DF송신, 어플리케이션 초기화, 어플리케이션 모드제어, 어플리케이션 제어, 표준시간 설정 등이다.
- 내용코드(CC; Content Code)는 데이터필드 아키텍처에서 수신주소를 지정하지 않고 송신측 서브시스템의 데이터 내용을 표시한 것이다. 송신측 서브시스템은 CC를 부가한 데이터를 데이터필드에 송신한다.

(6) 핵심서브시스템 데이터베이스 설계

- 핵심서브시스템 데이터베이스 설계는 각 열차 데이터베이스와 신호기, 선로 등의 데이터베이스를 설계하였다.
- 상세한 테이블 명은 알람테이블, 입력테이블, 출력테이블, 결합 스케줄 테이블, 신호기 테이블, 역 테이블, 선로전환기 테이블, 심벌테이블, 궤도회로 테이블, 진로테이블, 열차테이블 등이다.

다. 자율분산 노선관리시스템

(1) 라인모니터링 서브시스템

- 라인모니터링 서브시스템은 라인 전체 혹은 지정된 역에 대한 신호설비 상태정보 및 열차 운행상황을 기호 형태로 운영자에게 표시하고 운영자가 본선 열차운행계획을 관리할 수 있도록 GUI(Graphic User Interface)를 지원하는 장치이다.
- 라인모니터링 서브시스템의 역 선형정보 화면은 그래픽 툴을 이용하도록 하

며, 향후 역 선형정보가 수정되더라도 그래픽 툴을 이용한 역 화면 편집을 수행함으로써 프로그램의 변경 없이 새로운 역 선형정보를 표시할 수 있다.

(2) 라인경합관리 서브시스템

○ 라인경합관리 서브시스템(GCDRS; Global Conflict Detection & Resolution Subsystem)은 복수개의 역을 포함하는 서브네트워크의 열차운행관리 최적화를 위한 시스템으로 노선의 열차스케줄과 운행 중인 열차운행상태정보로부터 향후 일정한 시간 내에 발생할 경합을 예측하고 이를 해소한 후 열차지연을 최소화하는 라인변경스케줄을 라인정보 서브시스템으로 전송하는 역할을 한다.

○ 라인열차스케줄 정의 : 라인정보 서브시스템에 있는 전체 본선스케줄에서 현재 시각부터 일정시간 이후까지의 본선스케줄을 가져와서 GCDRS에서는 라인본선스케줄을 만들고 도착/출발 실적 정보 및 실적 역간이동시간을 반영하여 라인예측스케줄을 생성한다. 생성된 라인예측스케줄에 열차 간 시격유지 및 열차출발시간 제약을 만족시키지 못하여 경합이 검지되었을 경우 전체열차지연을 최소화 하도록 하는 라인변경스케줄을 생성한다.

○ 세부기능정의

- 메시지 송수신 : 라인정보 서브시스템 및 열차모니터링 서브시스템으로부터 본선스케줄 및 열차순서변경정보를 송수신한다.
- 경보/이벤트관리 : 라인경합관리 운영 중에 발생하는 각종 경보 및 이벤트를 기록하고 관리한다.
- 라인예측스케줄 관리 : 열차실적 도착 및 출발정보가 있으면, 실적 역간이동시간을 계산하여 다음 역 및 이후 역에 열차 출·도착시간을 예측한다.
- 라인경합검지 : 현재시간에서 30분~ 1시간 이내에 있는 라인예측스케줄에 대해서 열차의 경합상황이 있는지 검사한다. 라인경합이 발생한 첫 번째 경합에 대해서 경합정보를 생성하여 MMI에 표시한다.
- 라인경합해소 : 라인경합검지 정보가 있으면, 라인경합해소 알고리즘을 이용해서 경합해소를 실행한다.
- 라인변경스케줄전송 : 라인경합해소 결과로 만들어진 라인변경스케줄을 전송하기 전에, 라인관리자의 승인을 요청하고 승인된 변경스케줄을 라

인정보 서브시스템으로 전송한다.

○ 라인경합관리 서브시스템의 프로세스 정의

- 메시지 프로세스 : 데이터베이스 스캔, 메시지 송/수신 처리
- 경합관리 : 열차실적, 스케줄 관리, 라인경합검지, 라인경합해소
- 사용자 인터페이스 : 예측스케줄 수동변경 및 변경스케줄 승인 등의 사용자 조작
- 라인경보관리 : 통신오류 프로세스다운 및 스케줄 오류 등을 알람

○ 라인경합 검지 및 해소 수리모형 개발

라인경합관리 서브시스템에서 적용되는 열차경합해소문제는 이진(Binary)변수와 연속(continuous)변수가 혼합된 혼합정수계획법(Mixed Integer Programming)으

$$\min \sum_{p\in P_i}\sum_{i\in I_p} c_i (D_p^i - d_p^i) + w_i E_p^i$$

s.t.

$$(A_p^i - a_p^i) \le E_p^i, \quad \forall p\in P, \ \forall i\in I_p$$

$$-(A_p^i - a_p^i) \le E_p^i, \quad \forall p\in P, \ \forall i\in I_p$$

$$\sum_{k=1}^{n_p} Z_{p,k}^i = 1, \quad \forall p\in P, \ \forall i\in I_p$$

$$M(1-Z_{p,k}^i) + M(1-Z_{p,k}^j) + \begin{cases} M(1-X_p^{ij}) + D_p^i - D_p^j \ge h_r \\ M(1-X_p^{ij}) + A_p^i - A_p^j \ge h_r \\ MX_p^{ij} + D_p^i - D_p^j \ge h_r \\ MX_p^{ij} + D_p^i - D_p^j \ge h_r \end{cases}$$
$$\forall p=(q,r)\in P, \ k=1,\cdots,n_p, \ \forall i,j\in I_p$$

$$M(1-X_p^{ij}) + D_p^i - D_p^j \ge h_r, \quad \forall p=(q,r)\in P, \ \forall i,j\in I_p$$
$$MX_p^{ij} + D_p^i - D_p^j \ge h_r, \quad \forall p=(q,r)\in P, \ \forall i,j\in I_p$$
$$M(1-Y_p^{ij}) + A_p^i - A_p^j \ge h_r, \quad \forall p=(q,r)\in P, \ \forall i,j\in I_p$$
$$MY_p^{ij} + D_p^i - D_p^j \ge h_r, \quad \forall p=(q,r)\in P, \ \forall i,j\in I_p$$

$$2 - Z_{p,k}^i - Z_{p,k}^j + (X_p^{ij} - Y_p^{ij}) \ge h_m,$$
$$2 - Z_{p,k}^i - Z_{p,k}^j - (X_p^{ij} - Y_p^{ij}) \ge h_m,$$
$$\forall p=(q,r)\in P, \ \forall i,j\in I_p, \ k=1,\cdots,n_p$$

$$D_p^i \ge b_p^i, \quad \forall p\in P, \ \forall i\in I_p$$

$$\underline{t}_p^i(ss)SS_i^p + \underline{t}_p^i(sp)SP_i^p + \underline{t}_p^i(ps)PS_i^p + \underline{t}_p^i(pp)PP_i^p \le A_p^i - D_p^i,$$
$$\overline{t}_p^i(ss)SS_i^p + \overline{t}_p^i(sp)SP_i^p + \overline{t}_p^i(ps)PS_i^p + \overline{t}_p^i(pp)PP_i^p \ge A_p^i - D_p^i,$$
$$\forall p=(q,r)\in P, \quad \forall i\in I_p$$

$$SS_p^i + SP_p^i + PS_p^i + PP_p^i = 1, \quad \forall p\in P, \ \forall i\in I_p$$

$$D_p^i - A_p^i \ge s_q^i,$$
$$(SS_p^i + SP_p^i) - 1 + \epsilon \le D_p^i - A_p^i \le M(SS_p^i + SP_p^i),$$
$$(SS_p^i + PS_p^i) - 1 + \epsilon \le D_p^i - A_p^i \le M(SS_p^i + PS_p^i),$$
$$\forall q=(p',p)\in Q, \ \forall i\in I_p, I_{p'}.$$

로 모형화 된다.

○ 라인경합해소 알고리듬 개발 : 경합해소 알고리듬은 그림 3과 같이 경합 발생

시 신속하게 경합해소를 위하여 각 구간(역-역)별로 문제를 분리하고, 이렇게 분리된 부문제(subproblem)를 휴리스틱 알고리듬을 적용하였다. 따라서 각 구간을 위상정렬(topological sort)하여, 각 subproblem을 순차적으로 풀어나가는 방법으로 알고리듬을 개발하였다.

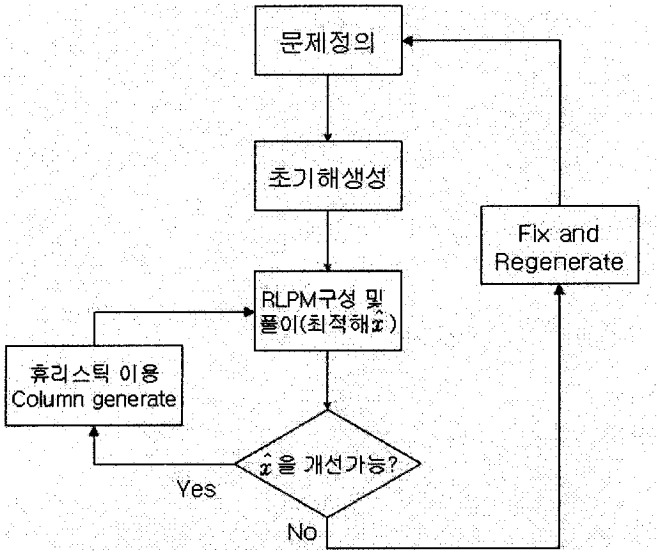


그림 3 알고리듬 전개 flow chart

(3) 라인정보 서브시스템

- 대형·고밀도 철도역 운행제어시스템 구축을 위해 라인정보 서브시스템이 구현되며, 라인정보시스템은 CTC의 운행관리 기능 중 일부인 열차스케줄 작성, 열차 그래프 기능을 해당 라인에 대해서 수행하도록 어플리케이션이 작성된다. 라인정보 서브시스템은 대형·고밀도운행 철도역 자동운행제어시스템 중에서 센터에 위치하게 되는 라인센터 시스템의 서브시스템이며 열차의 운행관리를 담당한다.
- 라인정보 서브시스템은 역 ASCS와 DF를 통해 연결되어 오프라인 열차스케줄의 전송 및 온라인 열차스케줄의 변경 정보를 전송하는 역할을 하게 된다.
- 라인정보 서브시스템에서 관리되는 열차 운행계획은 오프라인 및 온라인에 대한 본선스케줄로 제한이 되며, 역 입환 스케줄에 대한 관리는 수행하지 않는다.

3. 외부테스터 개발 및 시험

- 외부테스터(EXT; External Tester)는 자율분산시스템 개념을 도입하여 개발한 대형·고밀도운행 철도역 자동운행제어시스템을 관리자가 온라인 유지보수하거나 온라인 확장을 위한 테스트 도구로 개발되었음
- 실시간으로 운영되는 역 운행제어시스템은 대규모 지역에 설치되고 중지 없이 운영됨에 따라 고장 발생 빈도가 증가되어 실시간으로 온라인 유지보수와 온라인 확장에 대한 요구사항을 만족할 수 있는 외부테스터 개발이 요구됨
- 외부테스터 개발을 위한 요구사항은 UML 개발방법론에 의한 기능적 요구사항과 비 기능적 요구사항을 도출하였음
- 요구사항에 따라 도출된 외부테스터의 핵심기능은 출력 제한시간 검사 기능, 버전업 검사 기능, 스케줄 송수신 검사 기능, 경로제어 테스트 기능임
- 출력제한시간 검사 기능은 데이터필드를 통해 연결된 시스템에서 출력되는 데이터를 외부테스터에서 수신하여 대상시스템의 데이터 출력 제한시간을 초과하지 않고 출력하는지 테스트 함
- 출력제한시간 테스트 결과는 종합연계시험센터에서 수행하였으며 왕십리역을 대상으로 데이터필드에 연결된 외부테스터가 왕십리역에서 출력되는 역표시정보를 수신하여 테스트 하였으며 기준시각을 초과하는 데이터 없이 성공적으로 수행하였음

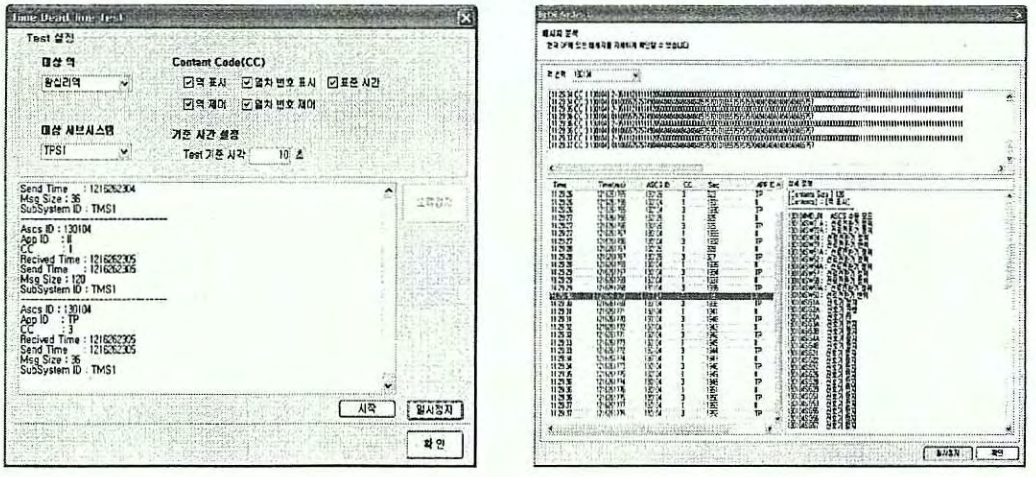


그림 4 출력제한시간 테스트 결과

o 버전업 검사 기능은 사용자의 요구사항을 반영하여 시스템 업그레이드 할 경우 이전버전과 신 버전에서 출력되는 데이터를 외부테스터에서 수신하여 신 버전 시스템에 오류가 있는지 판단하는 기능임

o 테스트 결과는 외부테스터에서 중량역 구버전 RTPS와 신버전 RTPS에서 출력되는 데이터를 수신하여 두 정보를 그림 5와 같이 수신하여 두 정보가 동일함을 테스트하였음

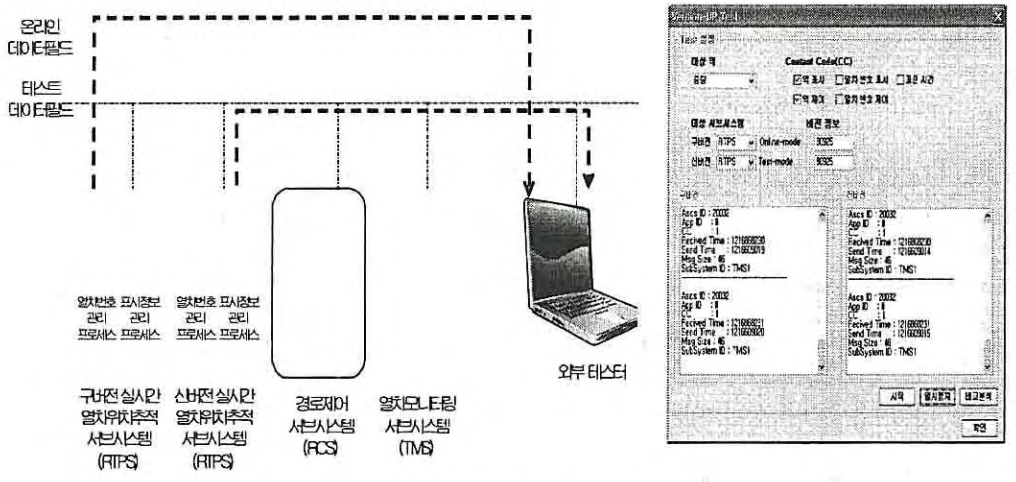


그림 5 버전업 테스트 결과

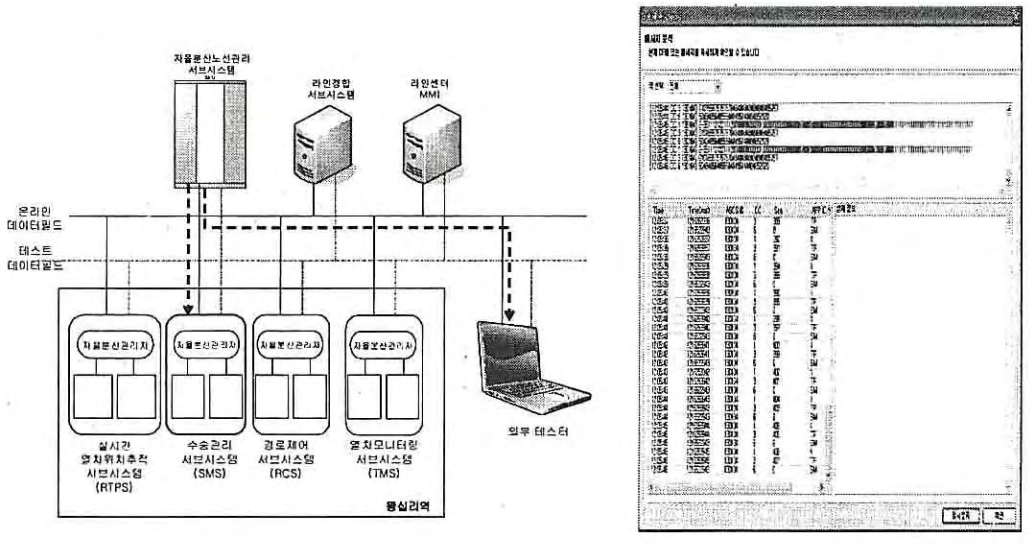


그림 6 스케줄 송수신 테스트 결과

- 스케줄 송수신 검사 기능은 센터에서 역으로 전송되는 스케줄은 용량이 커서 누락 없이 역으로 전송되는지를 검사하는 기능임
- 스케줄 송수신 테스트 결과는 자율분산 노선관리 서브시스템에서 송출된 정보를 수송관리 시스템에서 수신하는 데이터를 외부테스터에서 지속적으로 수신하여 데이터의 누락이 발생하지 않은 것을 확인하였음
- 경로제어 검사 기능은 경로제어서브시스템에서 진로 설정을 할 때 자동진로제어와 수동진로제어의 두 가지 방식을 테스트 할 수 있는 기능임
- 경로제어 검사 결과는 종합연계시험센터의 중랑역을 대상으로 테스트 하였으며 수동진로제어에 대해 테스트 하였으며, 사용자가 수동진로를 설정하였고 설정된 진로정보가 RCS에서 송신되고 그 송신된 정보를 RTPS에서 하부 장치 쪽으로 송출한 데이터가 동일함을 테스트하였음

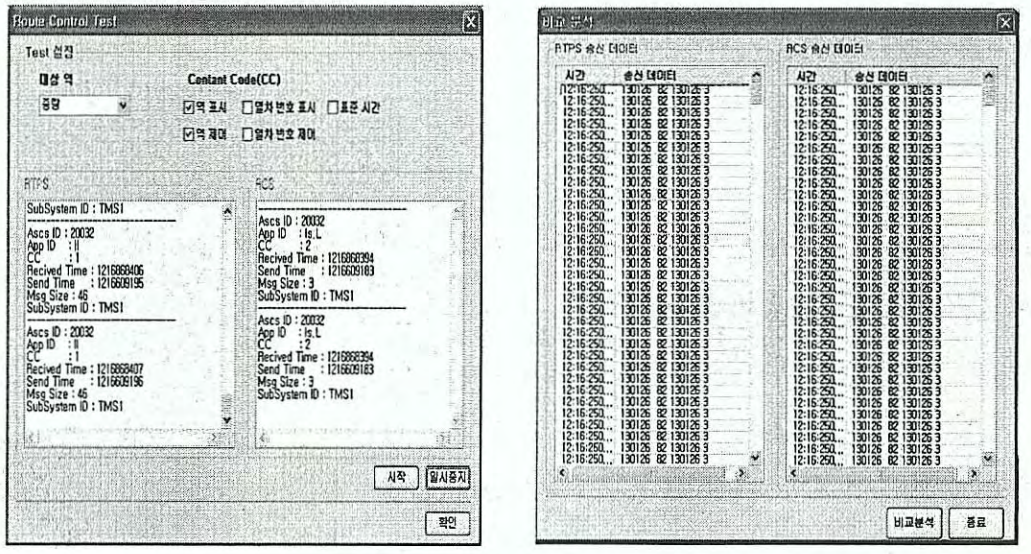


그림 7 경로제어 테스트 결과

4. 종합연계시험센터 구축 및 시험

○ 종합연계시험센터의 물리적 구성과 설치 사진은 그림 8, 그림 9와 같다.

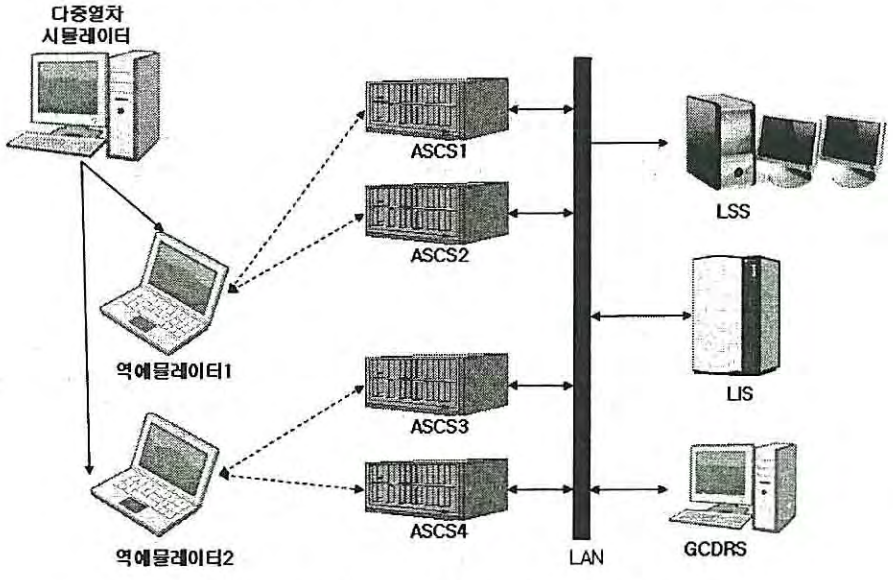


그림 8 종합연계시험센터의 물리적 구성

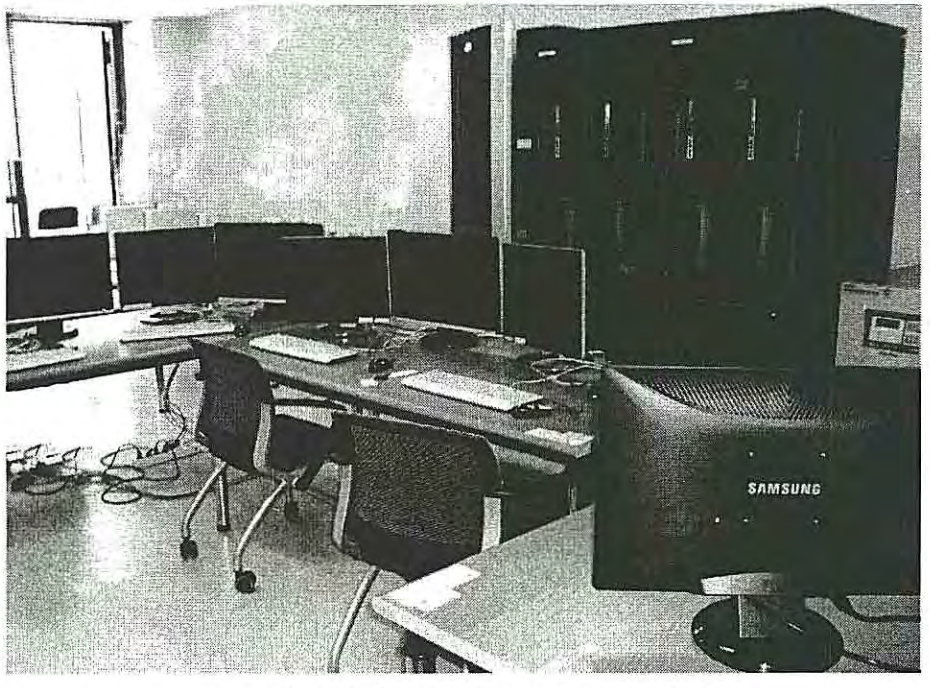


그림 9 종합연계시험센터 설치 사진

○ 종합연계시험센터의 구성은 자율분산 역제어시스템 4대와 열차운행의 상황을 모사하는 다중열차운행시뮬레이터와 역의 신호기와 상태정보를 모사하는 역

시뮬레이터, 라인을 운행하는 열차간의 경합을 해소하는 라인경합관리 서브시스템(GCDRS), 라인모니터링 서브시스템, 라인정보 서브시스템으로 구성된다.

- 종합연계시험센터 구축에 따른 시스템 기능 시험은 다중열차운행시뮬레이터와 역 에뮬레이터 인터페이스 시험을 수행하였으며 다중열차운행시뮬레이터의 자율분산관리자의 구동여부와 다중열차운행시뮬레이터의 구동 및 열차편집 기능 열차운행 및 정보처리 상황을 성공적으로 테스트 하였다.
- 라인정보시스템은 본선스케줄 전송과 수송관리 시스템에서의 스케줄 수신 상황을 테스트 하였고 온라인 스케줄의 변경정보 전송여부를 테스트 하였다.
- 각각의 자율분산 역제어시스템은 역 에뮬레이터와 다중열차운행시뮬레이터 등을 활용하여 열차 역 진입, 열차 주행, 역 진입예정열차 표시 상황 확인, 자동 및 수동 진로제어를 테스트 하였다.
- 종합연계시험센터를 활용하여 테스트 수행은 열차추적 서브시스템을 활용한 테스트, 경로제어 서브시스템의 주요 기능 테스트, 열차 모니터링 서브시스템을 기능 테스트를 성공적으로 수행하였다.

- 홈 리스케줄링 자체 기능 테스트는 홈 리스케줄링 프로세스와 동일 ASCS를 구성하는 다른 서브시스템(RTPS, RCS, SMS) 및 프로세스간의 연동 시험, 인접역 및 라인경합관리와의 연동이 되지 않는 상태에서 수행할 수 있는 홈 리스케줄링 경합 검지 및 해소 기능 테스트를 수행하였다.

  - 고장발생으로 인한 진로 사용 불가 검지 및 해소 : 궤도, 신호기, 선로전환기 등의 연동진로를 구성하는 Elements에 고장이 발생되어 고장정보가 RTPS를 통해 현장에서 올라오면 홈 리스케줄링에서는 이를 확인한 후



그림 10 궤도고장 경합검지 기능

사용불능으로 설정된 진로를 사용 예정된 열차스케줄이 있다면 열차번호, 도착/출발 진로여부, 진로명을 사용자에게 알람하여 준다. 이때, 사용자가 이를 우회할 수 있는 여러 대안 진로들을 입력하여 주면 홈 리스케줄링에서는 여러 대안 진로 중 홈 리스케줄링 대상열차들의 지연시간을 최소화하는 최적의 진로를 선택하여 변경 스케줄을 생성한다.

- 궤도 점유순서 불일치 검지 및 해소 : 역 운영자의 열차 출발 순서 변경 및 결합스케줄 변경으로 인하여 진로상의 열차가 점유하는 최종의 궤도의 도착순서와 출발순서를 비교하여 불일치한 열차스케줄이 있다면 경합이 발생된 열차번호들과 발생 최종 궤도를 사용자에게 알람 하여 준다. 이때, 사용자가 최종궤도의 점유순서를 일치 하도록 예측스케줄을 조정 요구하여 주면 홈 리스케줄링에서는 조정 요구된 스케줄을 바탕으로 홈 리스케줄링 대상열차들의 지연시간을 최소화하는 최적의 시간을 선택하여 변경 스케줄을 생성한다.

o 홈 리스케줄링 인접역 연동 기능 테스트는 홈 리스케줄링 프로세스와 인접역 ASCS 간에 연동이 되는 상태에서 수행할 수 있는 홈 리스케줄링 경합 검지 및 해소 기능 테스트를 수행하였다.

- 인접역 지연발생으로 인한 검지 및 해소 : 인접역에서 출발지연이 발생된 경우 해당 역 홈 리스케줄링에서는 인접역 RTPS로부터 생성된 열차 지연출발 정보를 확인한 후 지연발생 열차의 해당역 도착 시간을 재 예측한다. 재 예측된 예측스케줄과 기본스케줄과의 전체 지연발생 시간이 사용자가 지정한 한도시간을 초과할 경우 지연을 검지하여 전체지연발생

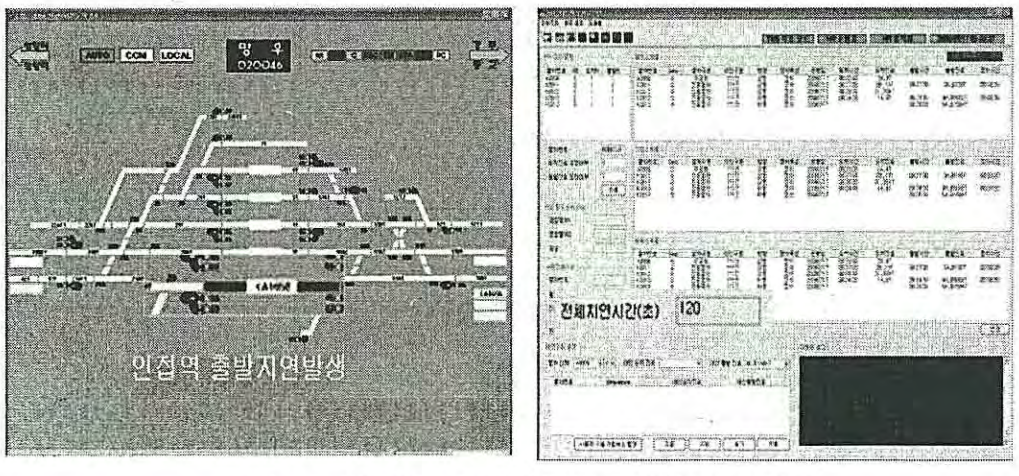


그림 11 인접역 지연발생으로 인한 경합검지

시간을 알람 한다. 이때, 홈 리스케줄링은 전체지연시간을 최소화 하는 변경 스케줄을 생성한다.

- 도착순서불일치 경합검지 및 해소 : 열차 예측 도착시간 변경으로 인하여 기본스케줄상의 상·하행 라인별 열차 도착순서와 예측스케줄상의 열차 도착순서가 일치 하지 않을 때 경합이 발생된 열차의 열차번호, 상·하행 정보, 라인정보를 사용자에게 알람 하여 준다. 이때, 사용자가 수동으로 예측스케줄을 조정하지 않으면 기본스케줄상의 도착 순서에 맞추어서 예측스케줄상의 열차 도착 및 출발 시간을 최적으로 조정한다.

o 홈 리스케줄링 프로세스와 라인경합관리 연동 테스트는 자율분산제어체계에서 계층적 구조로 설계되어있는 경합검지 및 해소기능을 하는 홈 리스케줄링과 라인경합관리간의 역할분담 및 협조설계에 대한 기능 테스트를 수행하였다.

- 홈 리스케줄링 출발순서 변경제안 테스트 : 출발순서를 고정시키는 경우 본선열차의 열차 출발 순서는 상하행, 라인별로 기존의 스케줄을 그대로 따르게 된다. 그러므로 라인경합관리로 열차순서 변경제안이 발생할 수 없다. 위와 반대로 출발순서를 변경가능 시키는 경우 홈 리스케줄링에서 역내 지연을 최소화하는 방향으로 본선열차의 열차 출발 순서가 상하행, 라인별로 기존의 스케줄에서 변경이 발생할 수 있게 된다. 이때, 순서변경 제안 정보를 라인경합관리로 전송하여 라인관리자는 이 정보를 바탕으로 전체 라인의 열차 운영을 변경한다.

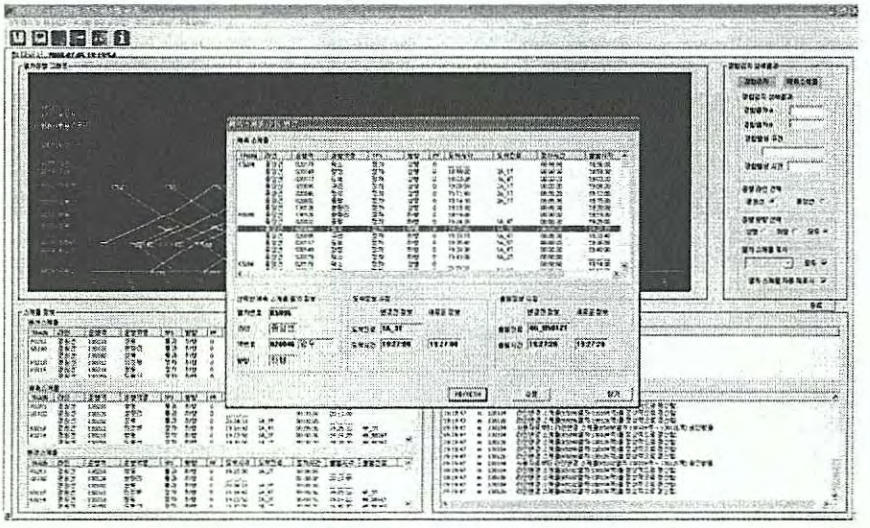


그림 12 출발순서변경 발생 경합해소

5. 현장시험시스템 구축 및 시험

가. 현장 시험시스템 구축

o 현장 시험 역은 청주역과 오근장역 오송역으로 선정되어 각 역에 자율분산 역제어시스템(ASCS)을 각 1대씩 구축하였으며 각 역간의 통신라인을 광케이블로 연결하여 구축하였다. 3개 역간 통신라인 구축은 그림 13과 같다.

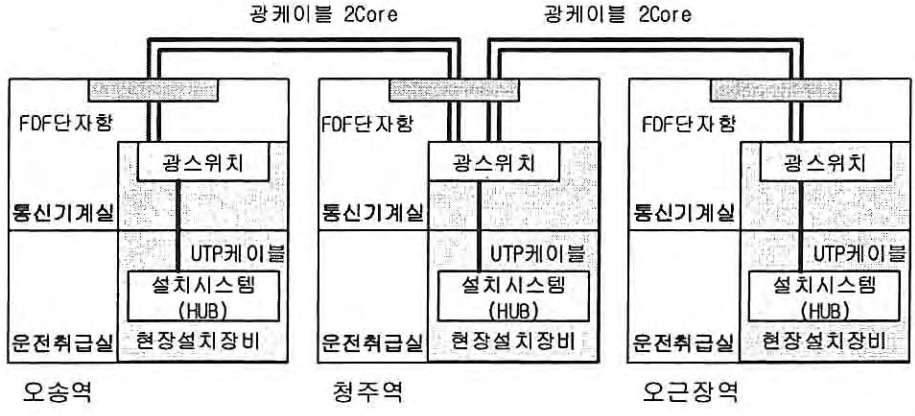


그림 13 역간 통신라인 구축

o 현장에서의 시스템 설치는 CTC에서 사용 중인 통신장치인 LDTS를 사용하였으며, LDTS와 전송속도는 9,600bps이다. 각 역간 시스템이 연결되는 방식은 그림 14와 같다.

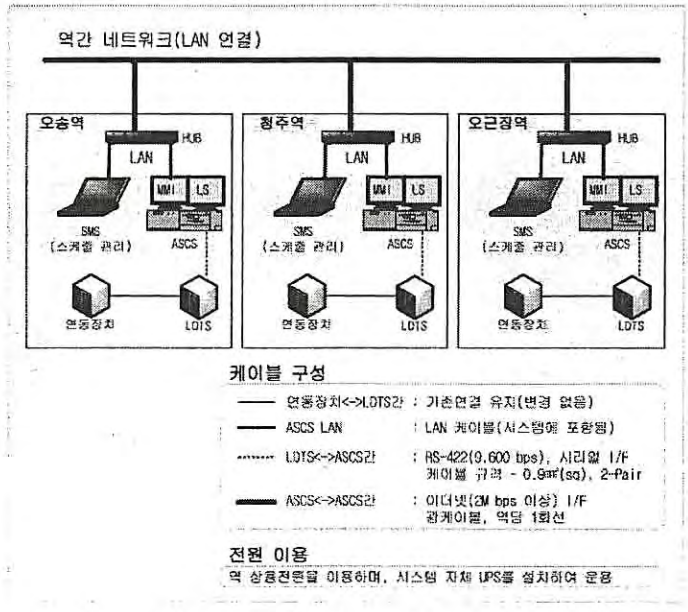


그림 14 현장 설치 시스템 구성도

o 시험을 위해 자율분산 역제어시스템 설비는 19" 표준랙이 각역에 설치되었고 랙 설비 내에 실시간 열차위치추적 서브시스템, 경로제어 서브시스템, 열차모니터링 서브시스템, 라인정보 서브시스템 등 4개의 서브시스템이 설치되었다. 그림 15는 현장역에 설치된 역제어시스템의 모습이다.

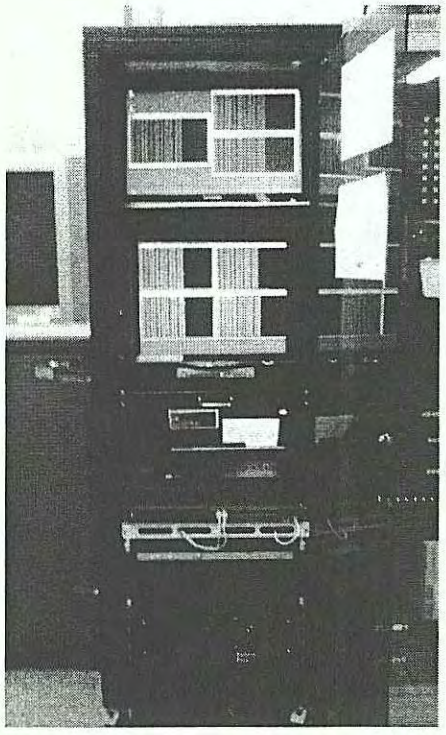


그림 15 현장역에 설치된
역제어시스템

나. 자율분산 속성 시험 및 온라인 확장시험

o 자율분산 역제어시스템의 자율 제어와 자율 협조 시험을 하였으며 자율 제어 시험은 인접역인 오송역 고장상태에서 영향 없이 청주역이 정상적으로 가동되는 상태를 시험하였으며, 자율 협조시험은 오근장역을 기능 불능상태로 설정하고 청주역과 오송역간의 협조성이 유지되어 이상 없이 구동되는 지를 테스트하였다.

o 온라인 확장시험은 단독역 상황에서 ASCS와 LDTS간의 연동시험, 열차추적 시험, 진로제어 시험을 성공적으로 수행하였고 인접역의 역시스템을 확장하는 단계의 시험에서는 인접역인 오송역을 온라인모드로 설정하고 확장하는 청주

역을 테스트 모드로 설정하여 오송역의 온라인 데이터를 청주역에서 수신하여 인접역 정보의 수신확인과 인접역 정보의 수신을 통한 진로제어가 정상적으로 이루어진 것을 확인 하였다. 그림 16은 인접역의 온라인데이터와 자기역의 테스트 데이터가 외부테스터를 통해 수신되는 것이며 그림 17은 청주역의 인접역인 오송역의 열차번호 정보를 수신한 것을 확인한 내용이다.

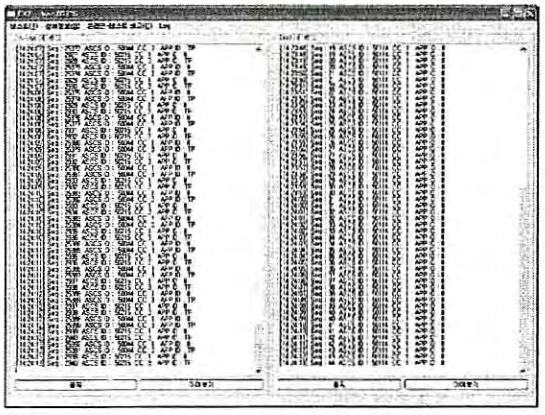


그림 16 데이터 수신확인

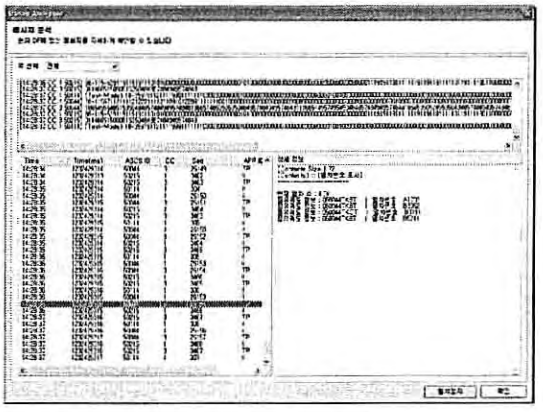


그림 17 오송역 열차출발정보 확인

다. 홈 리스케줄링 시험

o 홈 리스케줄링의 현장시험은 홈 리스케줄링 프로세스가 ADM으로부터 열차위치표시정보, 역상태정보, TMS모드정보 등의 실제 현장데이터를 수신 받아 정상 동작하는지와 현장 입환을 스케줄화 하였을 때 최적 변경스케줄을 작성하는지에 대한 테스트 항목을 도출하고 도출된 테스트 절차에 맞게 시스템을 테스트 하였다.

o 인접역 열차표시정보 수신테스트는 ASCS의 DF를 통하여 전송되는 인접역의 열차 표시정보를 HR에서 TMS의 ADM을 통하여 수신하는지를 테스트하는 것으로써 ASCS의 콘솔 화면 혹은 HR PARS로그파일을 통하여 상하행열차에 대하여 수신여부를 확인하였다.

o HR대상열차 예측스케줄 업데이트는 수신 받은 열차 표시정보 중에서 해당역의 HR대상열차에 포함되어있는 열차의 스케줄을 인접역의 열차표시정보를 활용하여 새롭게 예측스케줄을 작성하는지를 테스트하는 것으로써 HR PROC 로그파일을 통하여 상하행 열차에 대하여 처리여부를 확인하였다.

```
[2009-01-28 17:36:43] [CC 수신정보] 역코드 : 050215 , 열차번호 : B3051_0, 열차속성 :
    0x55
[2009-01-28 17:36:43] B3051_0 열차속성 : 상행, 출발, 통과
[2009-01-28 17:36:43] B3051_0 열차 예측시간을 다시 계산
[2009-01-28 17:36:43] DB로부터 050114RT2A_7T 진로의 예측시간 구함 , 통과시간 : 60
    초, 정차시간 : 120 초
[2009-01-28 17:36:43] TPS 시간 : PP -> 600
[2009-01-28 17:36:43] TrainStateChange, TPS: 600, CURTIME: 173642
[2009-01-28 17:36:43] TPS 600, PassMove :60 , StopMove : 120, CurrentTime : 173642
[2009-01-28 17:36:43] TrainStateChange, dpt : 174622, arv : 174622 DB에 기록
[2009-01-28 17:36:43] CALL SP_PREDICSCH_UPDATE('B3051', 0, '20090128', '174622',
    '050114RT2A_7T', '174622', '050114RT3C_B1010T');
```

o TMS 모드정보 활용테스트는 TMS 수동모드일 때와 TMS 자동모드일 때를 구분하여 수행하였다. 먼저 TMS의 모드정보 및 모드변경정보를 HR에서 수신하여 처리하는지를 테스트하였으며 이를 HR PARS로그파일과 TMS-L/S(Line Station Screen)상의 모드정보와 HR-GUI상의 모드정보의 일치 여부로 확인하였다.

o 두 번째로는 TMS 모드가 수동일 경우 HR의 기본기능(HR대상열차선정, 기본/예측스케줄 생성)은 정상적으로 동작하나 경합검지 및 해소는 발생하지 않으며 TMS가 자동모드일 때만 열차경합을 검지하고 이를 해소하여 리스케줄링 처리가 수행되는 것은 HR MMI상의 이벤트 로그를 통해 확인하였다



그림 18 TMS 수동모드
경합처리 로그

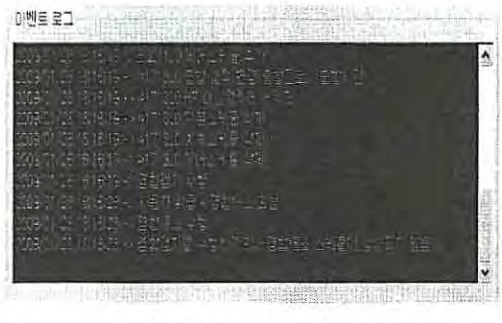


그림 19 TMS 자동모드
경합처리 로그

o HR은 본선스케줄 뿐만 아니라 입환 스케줄을 포함하여 경합을 검지하고 해소하는 기능을 가지고 있다. 하지만, 현장시험에서는 현재 역 운용원 및 수송원들이 따로 예측스케줄을 가지고 있지 않으므로 해당 역에서 수행되어지는 역 입환 상황을 입환 스케줄화 하여 시험을 수행하였다. 입환 스케줄 경합 검지 및 해소 테스트는 크게 입환이 하나의 입환 스케줄로 구성된 경우, 입환이 하나 이상의 입환 스케줄로 구성된 경우 두 가지로 시나리오를 구분하여 실시하였다.

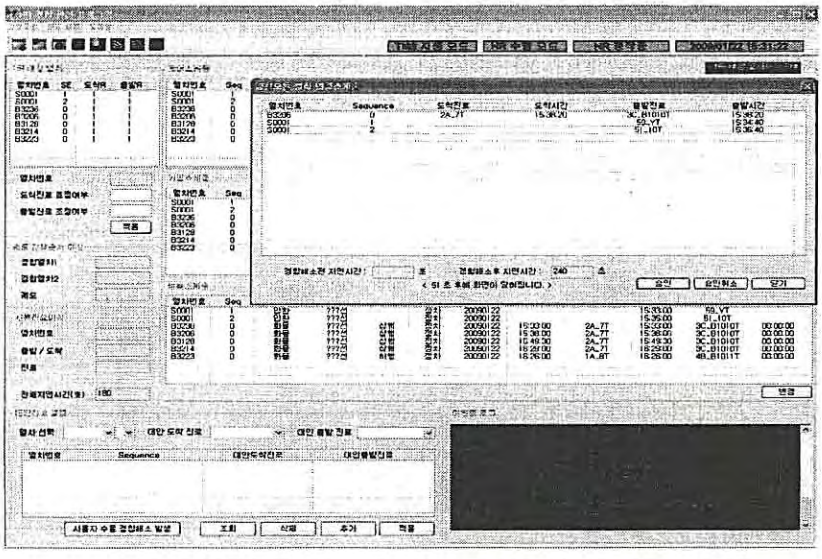


그림 20 입환 스케줄 하나 이상인 경우 리스케줄링

o 열차 출/도착 실적정보 활용테스트는 HR대상열차 중에서 실제 열차가 출발하고 도착함으로써 출/도착 시간 및 진로가 실적을 기록하면 이를 HR에서는 인지하고 해당열차를 일정시간 후에 HR 대상열차에서 삭제하는지를 테스트하는 것으로써 HR PROC로그파일 및 MMI를 통하여 수신여부를 확인하였다.

목 차

제출문 ····· I
요약문 ····· II
목 차 ····· XXIII
표 목차 ····· XXV
그림 목차 ····· XXVIII
용어 및 약어 ····· XXXVI

1장 연구개발과제의 개요 ····· 1
1절 연구배경 ····· 1
2절 연구목적 및 내용 ····· 2

2장 시스템 요구사항 분석 ····· 7
1절 국·내외 사례 조사 ····· 7
1. 국내 사례 ····· 7
2. 국외 현황 ····· 11
3. 국·내외 기술현황 분석 결과 ····· 17
2절 상시 로컬역 현황 조사 분석 ····· 19
1. 상시로컬 운영 현황 ····· 19
3절 시스템 요구사항 분석 ····· 25
1. 현장조사에 따른 요구사항 ····· 25
2. 자율분산형 시스템 요구사항 ····· 26

3장 대형·고밀도운행 철도역 자동운행제어 시스템 ····· 31
1절 자율분산 열차운행관리 시스템 구성 ····· 31
2절 자율분산 역제어시스템 ····· 32
1. 실시간 열차위치추적 서브시스템 ····· 32
2. 경로제어 서브시스템 ····· 48
3. 열차모니터링 서브시스템 ····· 54
4. 수송관리 서브시스템 ····· 73
5. 자율분산관리자 ····· 84
6. 핵심서브시스템의 데이터베이스 ····· 106
3절 자율분산 노선관리시스템 ····· 116
1. 라인모니터링 서브시스템 ····· 116
2. 라인경합관리 서브시스템 ····· 127
3. 라인정보 서브시스템 ····· 153

4장 외부테스터 ····· 189
1절 외부테스터 상세설계 ····· 189
1. 외부테스터 개요 ····· 189
2. 외부테스터 요구사항 도출 ····· 189
3. 외부테스터의 핵심 기능 ····· 191
4. 사용자 인터페이스 ····· 193
2절 종합연계시험센터에서의 기능 시험 ····· 203
1. 종합연계시험센터 환경 ····· 203
2. 출력 제한시간 시험 ····· 204
3. 스케줄 수신 시험 ····· 207
4. 경로제어 시험 ····· 208
5. 버전업 시험 ····· 210

5장 종합연계시험센터 ····· 213
1절 종합연계시험센터 개발 ····· 213
1. 종합연계 시험계획 작성 ····· 213
2. 현장 실 데이터 DB 구축 ····· 219
3. CTC에뮬레이션 서버와 타 시스템간 I/F 개발 ····· 222
4. 에뮬레이터 보완 제작 ····· 225
5. CTC L/S 모니터링 콘솔 개발 ····· 234
2절 종합연계시험센터 설치 및 시험 ····· 236
1. 종합연계시험센터 설치 ····· 236
2. 핵심서브시스템 테스트 ····· 248

6장 현장시험시스템 구축 및 시험 ····· 275
1절 현장시험 시스템 구축 ····· 275
1. 현장 시범시스템 구축을 위한 사전 조사 ····· 275
2. 현장 설치 ····· 278
2절 현장 시험 ····· 281
1. 자율분산 시스템 속성 시험 ····· 281
2. 온라인 시스템 확장 시험 ····· 287
3. 홈 리스케줄링 시험 ····· 296

7장 결 론 ····· 307
1절 결론 ····· 307
2절 향후 연구계획 ····· 308

참고문헌 ····· 310

표목차

표 1-1 연차별 연구목표 및 연구내용 ····· 3
표 2-1 ATOS 구현에 따른 주요효과 ····· 16
표 2-2 CTC와 ATOS의 주요 차이점 비교 ····· 17
표 2-3 계층형 분산시스템과 자율분산 시스템과의 비교 ····· 29
표 3-1 RTPS 서브랙 구성(1개역 기준) ····· 47
표 3-2 RCS 서브랙 구성(1개역 기준) ····· 53
표 3-3 홈 리스케줄링 프로세스 기능 ····· 65
표 3-4 스케줄 관리 서브프로세스 설명 ····· 66
표 3-5 데이터 분석 서브프로세스 설명 ····· 67
표 3-6 사용자 인터페이스 설명 ····· 68
표 3-7 경보관리 서브프로세스 설명 ····· 69
표 3-8 경보문구 구성 ····· 70
표 3-9 이벤트의 종류 ····· 70
표 3-10 경보 종류 및 등급 ····· 71
표 3-11 시스템간 송수신 메시지 ····· 102
표 3-12 공통 데이터 헤더 ····· 102
표 3-13 RC 유즈케이스 다이어그램 ····· 103
표 3-14 스케줄 데이터 포맷 ····· 104
표 3-15 제어데이터 포맷 ····· 105
표 3-16 현장 표시정보 포맷 ····· 106
표 3-17 표준시각 정보 포맷 ····· 106
표 3-18 알람테이블 ····· 107
표 3-19 입력 테이블 ····· 108
표 3-20 출력 테이블 ····· 110
표 3-21 결합 스케줄 테이블 ····· 111
표 3-22 신호기 테이블 ····· 112
표 3-23 역 테이블 ····· 113
표 3-24 선로전환기 테이블 ····· 113
표 3-25 심볼 테이블 ····· 114
표 3-26 궤도회로 테이블 ····· 114
표 3-27 진로 테이블 ····· 115
표 3-28 열차 테이블 ····· 115
표 3-29 라인경합관리 서브시스템 기능 ····· 134

표 3-30 GCDRS 메시지 관리 송신데이터 ····· 135
표 3-31 GCDRS 메시지 관리 수신데이터 ····· 135
표 3-32 GCDRS 라인경합관리 송신데이터 ····· 136
표 3-33 GCDRS 라인경합관리 수신데이터 ····· 136
표 3-34 GCDRS 사용자 인터페이스 송신데이터 ····· 137
표 3-35 GCDRS 사용자 인터페이스 수신데이터 ····· 138
표 3-36 GCDRS 라인경보관리 송신데이터 ····· 139
표 3-37 GCDRS 라인경보관리 수신데이터 ····· 139
표 3-38 경보문구 구성 ····· 139
표 3-39 경보 종류 및 등급 ····· 140
표 3-40 이벤트의 종류 ····· 141
표 3-41 스케줄관리 클래스에 대한 속성 정의 ····· 181
표 3-42 열차스케줄 클래스에 대한 속성 정의 ····· 181
표 3-43 역스케줄 클래스에 대한 속성 정의 ····· 182
표 3-44 진로패턴 클래스에 대한 속성 정의 ····· 182
표 3-45 타임 테이블 클래스에 대한 속성 정의 ····· 183
표 3-46 타임테이블 역별 클래스에 대한 속성 정의 ····· 183
표 4-1 외부테스터 요구사항 ····· 190
표 5-1 역 에뮬레이터 컴포넌트 정의 ····· 229
표 5-2 궤도회로 정보 송신 ····· 231
표 5-3 선로전환기 정보 송신 ····· 231
표 5-4 신호기 정보 송신 ····· 232
표 5-5 기계실 설정 ····· 232
표 5-6 역 모드 설정 ····· 233
표 5-7 역 모드 설정 ····· 233
표 5-8 폐색 표시 설정 ····· 233
표 5-9 자체기능시험 테스트 스케줄 ····· 253
표 5-10 궤도고장 경합검지 시험절차 및 결과 ····· 256
표 5-11 선로전환기고장 경합검지 시험절차 및 결과 ····· 257
표 5-12 신호기고장 경합검지 시험절차 및 결과 ····· 258
표 5-13 궤도 점유 순서 이상검지 시험절차 및 결과 ····· 259
표 5-14 궤도고장 경합해소 시험절차 및 결과 ····· 260
표 5-15 선로전환기 고장 경합해소 시험절차 및 결과 ····· 261
표 5-16 궤도고장 경합해소 시험절차 및 결과 ····· 262
표 5-17 성북 ASCS 본선스케줄 ····· 263
표 5-18 중랑 ASCS 본선스케줄 ····· 264
표 5-19 왕십리 ASCS 본선스케줄 ····· 264

표 5-20 인접역 지연발생 경합 시험절차 및 결과 ····· 267
표 5-21 도착순서불일치 경합검지 시험절차 및 결과 ····· 268
표 5-22 인접역 지연 발생 경합해소 시험절차 및 결과 ····· 269
표 5-23 도착순서불일치 경합검지 시험절차 및 결과 ····· 270
표 5-24 HR 출발순서 고정 시험절차 및 결과 ····· 272
표 5-25 HR 출발순서 변경 시험절차 및 결과 ····· 273
표 6-1 입환 상황 입환 스케줄화 ····· 302

그림목차

그림 1 자율분산 열차운행관리 시스템 구성도 ····· 4
그림 2 자율분산 열차운행관리 하드웨어 구성도 ····· 5
그림 3 알고리듬 전개 flow chart ····· 10
그림 4 출력제한시간 테스트 결과 ····· 11
그림 5 버전업 테스트 결과 ····· 12
그림 6 스케줄 송수신 테스트 결과 ····· 12
그림 7 경로제어 테스트 결과 ····· 13
그림 8 종합연계시험센터의 물리적 구성 ····· 14
그림 9 종합연계시험센터 설치 사진 ····· 14
그림 10 궤도고장 경합검지 기능 ····· 15
그림 11 인접역 지연발생으로 인한 경합검지 ····· 17
그림 12 출발순서변경 발생 경합해소 ····· 17
그림 13 역간 통신라인 구축 ····· 18
그림 14 현장 설치 시스템 구성도 ····· 18
그림 15 현장역에 설치된 역제어시스템 ····· 19
그림 16 데이터 수신확인 ····· 20
그림 17 오송역 열차출발정보 확인 ····· 20
그림 18 TMS 수동모드 ····· 21
그림 19 TMS 자동모드 ····· 21
그림 20 입환 스케줄 하나 이상인 경우 리스케줄링 ····· 22
그림 2-1 통합 CTC 물리적 구성 ····· 8
그림 2-2 통합 CTC 논리적 구성 ····· 9
그림 2-3 열차운행관리시스템 계층 ····· 12
그림 2-4 ATOS 도입이전 대형역에서 열차운행제어 문제점 ····· 13
그림 2-5 도쿄지역의 ATOS 제어 구간 ····· 15
그림 2-6 ATOS 시스템 구성도 ····· 15
그림 2-7 여객 중심의 상시로컬역 ····· 19
그림 2-8 청량리역 조직 구성 ····· 20
그림 2-9 화물중심의 상시로컬역 ····· 21
그림 2-10 망우역 조직 구성 ····· 22
그림 2-11 데이터필드 아키텍처의 논리적 구성 ····· 28
그림 3-1 자율분산 열차운행관리 시스템 구성도 ····· 31
그림 3-2 자율분산 열차운행관리 하드웨어 구성도 ····· 32

그림 3-3 RTPS 유즈케이스 ····· 34
그림 3-4 열차추적 ····· 35
그림 3-5 열 번관리 ····· 36
그림 3-6 실적 생성 ····· 37
그림 3-7 열번제어 ····· 38
그림 3-8 TP 클래스 다이어그램 ····· 39
그림 3-9 ILI 유즈케이스 ····· 42
그림 3-10 현장표시 ····· 42
그림 3-11 현장제어 ····· 43
그림 3-12 ILI 클래스 다이어그램 ····· 44
그림 3-13 RC 유즈케이스 다이어그램 ····· 48
그림 3-14 자동진로 유즈케이스 ····· 50
그림 3-15 RC 클래스 다이어그램 ····· 51
그림 3-16 홈 리스케줄링 프로세스 기본설계 ····· 56
그림 3-17 HR 열차스케줄 전이도 ····· 56
그림 3-18 홈 리스케줄링 프로세스 정의 ····· 65
그림 3-19 스케줄 관리 서브프로세스 DF ····· 66
그림 3-20 데이터 분석 서브프로세스 DFD ····· 67
그림 3-21사용자 인터페이스 DFD ····· 68
그림 3-22 경보관리 서브프로세스 DFD ····· 69
그림 3-23 홈 리스케줄링 ASCS연동 ····· 72
그림 3-24 온라인 스케줄 조회 흐름도 ····· 74
그림 3-25 입환 스케줄 관리 흐름도 ····· 75
그림 3-26 본선스케줄 조회 흐름도 ····· 77
그림 3-27 실적 스케줄 조회 흐름도 ····· 77
그림 3-28 SM 프로그램 패키지 구조 ····· 78
그림 3-29 SM 프로그램 클래스 다이어그램 ····· 78
그림 3-30 SM 프로그램 전체 화면 ····· 79
그림 3-31 온라인 스케줄 전체 조회 대화 창 ····· 79
그림 3-32 입환 스케줄 날짜선택 ····· 80
그림 3-33 입환 스케줄 관리 대화 창 ····· 80
그림 3-34 입환 스케줄 표시 ····· 81
그림 3-35 스케줄 삭제 ····· 81
그림 3-36 입환 스케줄 작성 ····· 81
그림 3-37 입환 스케줄 연계 열번 자동 입력 ····· 82
그림 3-38 입환 스케줄 삭제 확인 ····· 82
그림 3-39 오프라인 본선스케줄 전체 조회 대화 창 ····· 83

그림 3-40 실적 스케줄 조회 대화 창 ····· 83
그림 3-41 데이터 필드간 송수신 ····· 84
그림 3-42 멀티 캐스트 그룹 ····· 85
그림 3-43 노드간 피어-투-피어 송수신 ····· 86
그림 3-44 APP제어 ····· 88
그림 3-45 ADM 유즈케이스 ····· 90
그림 3-46 APP제어 ····· 90
그림 3-47 DF 송신 ····· 91
그림 3-48 DF수신 ····· 92
그림 3-49 APP 구동 ····· 94
그림 3-50 ADM 클래스 다이어그램 ····· 94
그림 3-51 ASCS-ERD ····· 107
그림 3-52 라인모니터링 서브시스템의 구성 ····· 116
그림 3-53 라인모니터링의 자율분산관리자 ····· 119
그림 3-54 자율분산관리자에 의해 저장된 데이터 ····· 120
그림 3-55 라인모니터링 시험 대상역(오근장) ····· 120
그림 3-56 라인모니터링 시험 대상역(청주) ····· 121
그림 3-57 라인모니터링 시험 대상역(오송) ····· 121
그림 3-58 라인모니터링 시험 대상 라인(충북선:오송-오근장 구간) ····· 122
그림 3-59 라인모니터링 자료 흐름도 ····· 123
그림 3-60 오프라인 열차스케줄 ID 관리 ····· 123
그림 3-61 온라인 열차스케줄 ID 관리 ····· 124
그림 3-62 진입 예고열차 정리 ····· 124
그림 3-63 오프라인 열차스케줄 관리 ····· 125
그림 3-64 오프라인 열차스케줄 편집 ····· 125
그림 3-65 오프라인 열차스케줄 ID간 열차복사 ····· 126
그림 3-66 온라인 열차스케줄 관리 ····· 126
그림 3-67 라인경합관리 서브시스템 ····· 127
그림 3-68 라인경합관리 스케줄 전이도 ····· 128
그림 3-69 라인경합관리 서브시스템 ····· 133
그림 3-70 GCDRS 메시지관리 DFD ····· 134
그림 3-71 라인경합관리 DFD ····· 135
그림 3-72 GCDRS 사용자 인터페이스 DFD ····· 137
그림 3-73 GCDRS 라인경보관리 FLOW ····· 138
그림 3-74 알고리듬 전개 flow chart ····· 153
그림 3-75 라인정보 서브시스템 구성도 ····· 154
그림 3-76 역 단독 모드 구축 개념도 ····· 155

그림 3-77 단일역 구축 개념도 ····· 155
그림 3-78 인접역 확장 개념도 ····· 156
그림 3-79 자율분산 노선관리 확장 개념도 ····· 157
그림 3-80 기존 열차스케줄 관리 흐름도 ····· 158
그림 3-81 기존노선에 ADTMS가 도입되는 경우의 스케줄 관리 흐름도 ····· 159
그림 3-82 신규 노선에 ADTMS가 도입되는 경우의 스케줄 관리 흐름도 ····· 159
그림 3-83 라인정보 서브시스템의 스케줄 전송 흐름도 ····· 160
그림 3-84 라인정보 서브시스템과 ASCS간의 데이터 동기화 ····· 161
그림 3-85 자율분산시스템과의 병행운전을 위한 ····· 162
그림 3-86 기존 관제시스템의 연결 구조 ····· 164
그림 3-87 ADTMS도입에 따른 관제시스템의 연결 구조 변화 ····· 164
그림 3-88 ASCS와 라인센터의 Data Field 구성 ····· 165
그림 3-89 라인정보 서브시스템과 CTC간의 데이터 흐름(기능측면) ····· 166
그림 3-90 라인센터 시작 ····· 167
그림 3-91 익일 열차스케줄 로드 ····· 167
그림 3-92 열차운행계획 추가 ····· 168
그림 3-93 열차운행계획 삭제 ····· 168
그림 3-94 열차위치추적 표시 클래스 다이어그램 ····· 179
그림 3-95 < 오프라인 스케줄 작성 > 클래스 다이어그램 ····· 180
그림 3-96 열차스케줄 수신 클래스 다이어그램 ····· 184
그림 3-97 < 온라인 스케줄 수신 > 클래스 다이어그램 ····· 185
그림 3-98 운행실적관리 클래스 다이어그램 ····· 186
그림 3-99 열차스케줄 정보관리 클래스 다이어그램 ····· 187
그림 4-1 외부테스터 초기화면 ····· 194
그림 4-2 메시지 분석 창 ····· 195
그림 4-3 데이터필드 연결 화면 ····· 196
그림 4-4 출력 제한시간 검사 화면 ····· 197
그림 4-5 시간 분석 화면 ····· 198
그림 4-6 Version up 검사 화면 ····· 199
그림 4-7 버전업 검사 비교분석 화면 ····· 200
그림 4-8 스케줄 송수신 검사 화면 ····· 200
그림 4-9 데이터 누락 검사 화면 ····· 201
그림 4-10 경로제어 검사 화면 ····· 201
그림 4-11 경로제어 검사 비교 분석 화면 ····· 202
그림 4-12 종합연계시험센터 시스템 구성도 ····· 203
그림 4-13 종합연계시험센터 시험실 ····· 204
그림 4-14 출력 제한시간 시험 개요도 ····· 205

그림 4-15 주기적 통신 제한시험 화면 ····· 206
그림 4-16 역표시 데이터의 상세 내용 ····· 206
그림 4-17 스케줄 수신 시험 개요도 ····· 207
그림 4-18 수신된 스케줄 데이터의 상세 결과 ····· 208
그림 4-19 경로제어 시험 개요도 ····· 209
그림 4-20 수동진로 테스트 결과 ····· 210
그림 4-21 경로제어 송신데이터 비교 결과 ····· 210
그림 4-22 버전업 시험 개요도 ····· 210
그림 4-23 버전업 테스트 시험 결과 ····· 211
그림 5-1 테스트 프로세스 구성요소 ····· 213
그림 5-2 Altibase Admin TOOL ····· 220
그림 5-3 마이그레이션 된 구로CTC 데이터베이스 ····· 220
그림 5-4 현장 신호설비 상태정보 전송 ····· 222
그림 5-5 CTC에뮬레이션 서버의 스케줄 전송 ····· 223
그림 5-6 CTC에뮬레이션 서버의 프로세스 기동 흐름(변경 후) ····· 225
그림 5-7 MTMS 유즈케이스 다이어그램 ····· 227
그림 5-8 역 에뮬레이터 시스템 경계 ····· 228
그림 5-9 역 에뮬레이터 로직컬 뷰 ····· 229
그림 5-10 역 에뮬레이터 컴포넌트 뷰 ····· 229
그림 5-11 역 에뮬레이터 물리적 뷰 ····· 230
그림 5-12 역 에뮬레이터 전체 화면 ····· 230
그림 5-13 CTC L/S 모니터링-충북선 오송역 ····· 234
그림 5-14 CTC L/S 모니터링-충북선 청주역 ····· 235
그림 5-15 CTC L/S 모니터링-충북선 오송역 ····· 235
그림 5-16 종합연계시험센터의 논리적 구성 ····· 236
그림 5-17 종합연계시험센터의 물리적 구성 ····· 237
그림 5-18 종합연계시험센터 설치 사진(1) ····· 237
그림 5-19 종합연계시험센터 설치 사진(2) ····· 238
그림 5-20 종합연계시험센터 설치 사진(3) ····· 238
그림 5-21 종합연계시험센터의 데이터필드 구성 ····· 239
그림 5-22 다중열차운행시뮬레이터를 위한 ADM 구동 ····· 240
그림 5-23 다중열차운행시뮬레이터 구동 ····· 241
그림 5-24 다중열차운행시뮬레이터-열차편집 ····· 241
그림 5-25 SIM-열차운행시작 ····· 242
그림 5-26 SIM-열차주행 ····· 242
그림 5-27 역에뮬레이터 실행 ····· 242
그림 5-28 역에뮬레이터 정보처리 ····· 242

그림 5-29 LIS-자율분산관리자 시작 ····· 243
그림 5-30 LIS-스케줄관리자 시작 ····· 243
그림 5-31 SMS-오프라인스케줄수신 ····· 243
그림 5-32 LIS-오프라인스케줄 전송 ····· 243
그림 5-33 오프라인스케줄 수신완료 ····· 244
그림 5-34 SMS-수신된 본선스케줄표시 ····· 244
그림 5-35 LSS-온라인스케줄 관리 ····· 244
그림 5-36 LIS-온라인스케줄 변경 ····· 244
그림 5-37 LIS-온라인스케줄 변경정보 전송 ····· 244
그림 5-38 온라인스케줄 수신완료 ····· 244
그림 5-39 역에뮬레이터 구동 ····· 245
그림 5-40 역에뮬레이터 실행화면 ····· 245
그림 5-41 ASCS 열차 역 진입 ····· 245
그림 5-42 열차 주행 ····· 245
그림 5-43 역 진입예정열차 표시 ····· 245
그림 5-44 수동진로제어 ····· 245
그림 5-45 진입진로 구성 ····· 246
그림 5-46 열차 구내 진입 ····· 246
그림 5-47 역 도착실적 기록 ····· 246
그림 5-48 역 출발진로 설정 ····· 246
그림 5-49 역 장내 출발 ····· 246
그림 5-50 역 출발실적 기록 ····· 246
그림 5-51 라인센터 열차추적 ····· 247
그림 5-52 EXT를 이용한 모니터링 ····· 247
그림 5-53 ADTMS 테스트 로그 데이터(부분) ····· 248
그림 5-54 RTPS ILI-현장 신호설비 표시정보 기능-환경구성 ····· 248
그림 5-55 RTPS ILI-환경구성 기능-환경구성 ····· 249
그림 5-56 RTPS ILI-현장 제어메시지 전송 기능-환경구성 ····· 249
그림 5-57 RTPS 열차추적 기능-환경구성 ····· 249
그림 5-58 TMS 기능 테스트 환경 ····· 249
그림 5-59 자체테스트 시험환경 구성 ····· 253
그림 5-60 궤도고장 경합검지 기능 ····· 255
그림 5-61 선로전환기고장 경합검지 ····· 256
그림 5-62 신호기고장 경합검지 ····· 257
그림 5-63 궤도 점유 순서 이상검지 ····· 258
그림 5-64 궤도고장 경합해소 기능 ····· 259
그림 5-65 선로전환기 고장 경합해소 ····· 260

그림 5-66 인접역 연동 테스트 시험환경 구성 ····· 263
그림 5-67 인접역 지연발생으로 인한 경합검지 ····· 266
그림 5-68 도착순서불일치 경합검지 ····· 267
그림 5-69 인접역 지연 발생 경합해소 ····· 268
그림 5-70 라인경합관리 연동 테스트 시험환경 구성 ····· 271
그림 5-71 출발순서변경 발생 경합해소 ····· 273
그림 6-1 오송역 조작반 ····· 275
그림 6-2 오송역 선형표시 ····· 275
그림 6-3 오근장역 선형표시 ····· 276
그림 6-4 오근장역 LDTS ····· 276
그림 6-5 청주역 조작반 ····· 276
그림 6-6 청주역 KROIS ····· 276
그림 6-7 통신 랙 ····· 277
그림 6-8 통신 단자함 ····· 277
그림 6-9 역간 통신라인 구축 ····· 277
그림 6-10 현장 설치 시스템 구성도 ····· 278
그림 6-11 설치된 시험시스템 ····· 279
그림 6-12 오송역 설치 후 ····· 280
그림 6-13 오송역 LDTS 단자 연결 ····· 280
그림 6-14 오근장역 설치 후 ····· 280
그림 6-15 오근장역 운영 ····· 280
그림 6-16 청주역 설치 후 ····· 280
그림 6-17 청주역 운영 ····· 280
그림 6-18 청주역에서 오근장역, 오송역 정보수신 화면 ····· 282
그림 6-19 인접역 데이터 수신 불능 상태 확인 ····· 283
그림 6-20 진로제어 명령의 표시 ····· 283
그림 6-21 외부테스터에서의 진로명령 출력 확인 ····· 284
그림 6-22 자율협조 시험 상황 개요도 ····· 285
그림 6-23 청주역 외부테스터에서 오근장역의 정보 수신 ····· 285
그림 6-24 청주역에서 오근장역 정보가 수신되지 않는 상태 ····· 286
그림 6-25 청주역 열차 진로 설정 확인 ····· 286
그림 6-26 단독역 테스트 구성도 ····· 288
그림 6-27 LDTS 연동시험 결과 ····· 288
그림 6-28 현장 정보 수신에 의한 TMS 화면 표시 결과 ····· 289
그림 6-29 열차추적 시험 결과 ····· 289
그림 6-30 진로제어 시험 결과 ····· 290
그림 6-31 외부테스터를 활용한 제어정보출력 결과 확인 ····· 291

그림 6-32 온라인 확장 테스트 절차 ····· 292
그림 6-33 인접역 확장 시험 개념도 ····· 292
그림 6-34 온라인 확장을 위한 인접역 온라인 구동 확인 ····· 293
그림 6-35 자율분산관리자에서의 테스트모드 설정 ····· 293
그림 6-36 온라인데이터와 테스트데이터의 수신확인 ····· 294
그림 6-37 오송역 열차출발정보 확인 ····· 294
그림 6-38 역시스템 확장 단계의 열차진로제어 시험 ····· 295
그림 6-39 홈 리스케줄링 시험대상 영역 ····· 296
그림 6-40 TMS/HR 수동모드 일치 테스트 ····· 300
그림 6-41 TMS 수동모드 경합처리 로그 ····· 300
그림 6-42 TMS 자동모드 테스트 ····· 301
그림 6-43 TMS 자동모드 경합처리 로그 ····· 301
그림 6-44 입환 상황 예시 ····· 302
그림 6-45 입환스케줄 하나인 경우 리스케줄링 ····· 303
그림 6-46 입환스케줄 하나 이상인 경우 리스케줄링 ····· 304

용어 및 약어

ADM	Autonomous Decentralized Manager 자율분산관리자
ADTMS	Autonomous Decentralized Traffic Management System 대형 · 고밀도운행 철도역 자동운행제어시스템
APP	Application Program
ASCS	Autonomous Station Control System 자율분산 역제어시스템
ATOS	Autonomous decentralized Transport Operation control System 동경권 열차운행제어시스템
BIT	Built ib Tester
CC	Content Code
CDRS	Conflict Detection & Resolution System
CTC	Central Traffic Control(중앙 집중 제어)
DF	Data Field
DIA	Diagram
EXT	External Tester
GCDRS	Global Conflict Detection & Resolution Subsystem 라인경합관리 서브시스템
GUI	Graphic User Interface
HR	Home Rescheduling
IDF	Internal Data Field
ILI	Interlocking Interface
KROIS	Korea Rail Operation Information System
L/S	Line Station Screen
LDTS	Local Data Transmission System 역 정보 전송장치
LIS	Line Information System
LPM	Linear Programming Master Problem
MMI	Man Machine Interface
RCS	Route Control Subsystem 경로제어 서브시스템

RLPM	Restricted Linear Programming Master Problem
ROM	Read Only Memory
RTPS	Real-time Train Positioning Subsystem 실시간 열차위치추적 서브시스템
SMS	Shunting Management System
TMS	Train Monitoring Subsystem 열차모니터링 서브시스템
TPS	Train Performance Simulator
UDP	User Datagram Protocol

1장 연구개발과제의 개요

1절 연구배경

철도교통의 제어체계는 크게 열차제어(Train Control), 연동제어(Interlocking), 운행제어(Train Traffic Control)로 구분할 수 있다. 열차제어는 지상으로부터 운행중인 철도차량에 제동관련 제어에 필요한 정보를 지속적으로 주어 열차가 상황에 적절한 속도로 운행 할 수 있도록 하는 방식이며 연동 제어는 선로위에 다수의 열차가 동시에 움직이는 경우 선로상의 각 신호제어장치는 서로 간에 경합이 일어나지 않도록 작동하는 것이다. 본 연구에서도 개발하고 있는 운행 제어는 모든 열차가 정해진 시간에 진로를 따라 운행할 수 있도록 선로전환기를 사용하여 열차의 진로를 설정하는 것이다.

열차운행제어시스템은 계획된 시간표에 따라 운행되는 열차의 정시성 제공을 위한 철도시스템의 핵심부분으로 운행노선과 역들을 관리하며 특히 열차에 대한 역과 노선에 자원할당과 열차들의 경합 및 대기를 방지하는 역할을 수행한다.

국내 열차운행제어기술은 철도 역 현장의 신호설비를 중앙 집중 제어하여 효율을 높이는 열차집중제어장치(CTC 장치)로 발전하여 왔다. 기존의 열차집중제어장치는 마이크로프로세서 보드 기반의 시스템으로 전국 철도노선을 운행하는 열차 및 현장제어장치를 원격으로 감시 및 제어하는 역할을 수행하여 열차집중제어장치는 트래픽 관리보다 트래픽 제어시스템의 성격이 강했다.

일본의 열차운행제어기술은 국내와 같이 중앙에서 선로전환기를 원격으로 제어하는 방식으로 운영되었지만 역이 대형화 되고 입환 작업이 생기면서 프로그램에 의한 자동진로설정이 어려워졌다. 이에 동일본 철도에서는 동경지역 300여개 역을 대상으로 수송관리 시스템을 개발하여 도입하였다.

본 과제에서는 운행제어시스템이 대규모 지역을 관리하고 역 규모 등의 특성에 맞는 목적과 기능을 수행하며 전체 시스템 관점에서 하나의 시스템이 불가동한 상태에 있더라도 전체시스템의 영향을 최소화하여 운행할 수 있으며 지속적인 사용자 요구사항을 시스템에 반영하여 그 기능을 향상할 수 있는 온라인 확장 기술과 온라인 유지보수 기능 등의 요구사항을 반영한 운행제어시스템을 개발하였다.

2003년 6월부터 2008년 12월까지 5차년도로 개발된 대형·고밀도운행 철도역 자동운행제어시스템은 주관기관인 한국철도기술연구원과 협동연구기관인 (주)대아Ti가 공동으로 연구 개발을 수행하였다. 5차 년도에는 서울대학교에서 열차경합해소 알고리즘 개발에 위탁연구기관으로 참여하여 산·학·연 연구개발 체계로 구성되어 연구가 진행되어졌다.

연구개발 첫해에는 시스템 요구사항 분석과 자율 분산형 열차운행제어 연구와 자율 분산형 시스템 아키텍처 개발을 수행하였다. 2차 년도부터 4차 년도까지는 대형·고밀도운행 철도역 자동운행제어시스템에 포함되는 서브시스템 개발을 수행하였다. 이 시스템은 센터의 역할을 수행하여 스케줄과 경합관리를 총괄하는 라인모니터링 서브시스템, 라인경합관리 서브시스템, 라인정보 서브시스템이 개발되었으며 자율분산 역제어시스템은 핵심 서브시스템인 실시간 열차추적 서브시스템, 경로제어 서브시스템, 열차 모니터링 서브시스템이 개발되었고, 지원 서브시스템인 수송관리 서브시스템이 개발되었다.

2절 연구목적 및 내용

본 과제의 최종 목표는 열차운행제어 처리용량과 수송능력 및 안전도를 극대화하기 위하여 기존의 중앙 집중제어 방식의 열차운행제어시스템과 병행운전을 통하여 고밀도운행 및 역 자원 최적 활용이 가능한 철도역단위 열차운행제어시스템을 개발하는 것이다. 이는 철도형태의 다양화, 운영주체의 다양화, 철도네트워크의 증가로 인한 혼잡도 가중 및 선로와 역 용량의 부족현상을 소프트웨어적으로 개선할 수 있는 자율 분산형 차세대 열차운행제어시스템 기술 개발을 하려는 것이다.

이 보고서는 총 5차년도의 계획 중 5차년도 최종 연구보고서로 전체 연구목표 및 연구내용은 표 1-1과 같다.

표 1-1 연차별 연구목표 및 연구내용

구분	연구개발 목표	연구개발내용 및 범위
1차년도	시스템 요구사항 수집/분석, 시스템 아키텍처 개발	○수도권 철도망계획 및 직결/연계운용계획에 따른 요구사항 조사·분석 ○자율분산제어 기술 및 실시간 트래픽 최적화 기술 적용방안 연구 ○일본 철도통합연구소(RTRI)와 프랑스 철도운영사(SNCF)의 기존연구 및 기술개발내용 분석을 통한 요구사양 작성 ○국내 상시로컬제어역의 운행제어현황 및 조사를 통한 실제문제의 정확한 분석을 토대로 시스템 아키텍처 개발
2차년도	대형역 자동운행 제어시스템 상세설계	○ 자율분산형 열차운행제어 체계 상세 설계 ·협조논리 설계 ·자율분산관리자 설계 ·통신체계 및 프로토콜 설계 ○ 자율분산형 열차운행체계 시뮬레이터 개발 ·시뮬레이터 기능 및 시나리오 설계 ·자율분산 역 시뮬레이터 개발 ·이벤트 시뮬레이터 개발 ·역 에뮬레이터 개발 ·CTC 에뮬레이터 개발 ○자율분산형 열차운행제어 체계 시뮬레이션을 통한 효과 분석 ·자율분산 효과 검증을 위한 시나리오 설계 ·시나리오별 효과분석 시험 ·분석시험을 통한 보안책 도출
3차년도	대형역 자동운행 제어 핵심시스템 개발 및 시험	○핵심서브시스템 상세 설계 ·실시간 열차위치추적 서브시스템 ·경로제어 서브시스템 ·열차모니터링 서브시스템 ○시험대상역별 프로토타입 제작 ·핵심서브시스템 프로토타입 제작 ·S/W, H/W 인티그레이션

		○ 홈 리스케줄링 프로세스 설계 및 알고리즘 개발 · 홈 리스케줄링 업무 절차 정형화 · 분석적 접근방법을 통한 홈 리스케줄링 절차 개선 방안 도출 · 역 자율성에 대한 역과 센터의 운영범위 및 역할 정립 · 홈 리스케줄링 프로세스 기본설계 및 알고리즘 개발 ○ 핵심 서브시스템 테스트 · 시뮬레이터와 핵심서브시스템 통합 · 시뮬레이터를 활용한 테스트 시나리오 작성 · 핵심서브시스템 시제품 테스트
4차년도	대형역 자동운행 제어 서브시스템 개발 및 시험	○ 서브시스템 개발 및 시험 · 수송관리, 라인정보, 라인모니터링 서브시스템 · 센터 연동모드 테스트 시나리오 설계 · 인티그레이션 테스트 ○ 핵심서브시스템 개발 및 시험 · 2개역 대상 핵심서브시스템 개발 및 제작 · 인접역 연동모드 테스트 시나리오 설계 · 인티그레이션 테스트 · 실시간 DATA 처리를 위한 기술(OMG의 DDS) 검토 ○ 통합관제실 시뮬레이터 개발 · CTC서버 기능 구현 · 스케줄서버 기능 구현 · 통신서버 기능 구현 ○ 홈 리스케줄링 개발 · 홈 리스케줄링 상세설계 · 홈 리스케줄링 개발 및 테스트 ○ 라인경합관리 및 자율분산속성 테스트 장비 구축 · 라인경합관리 개념설계 · 외부테스트 시스템 개발
5차년도	시범시스템 구축 및 시험/검증	○ 핵심서브시스템 제작 및 시험 · 왕십리, 성북역 핵심서브시스템 제작 · 홈 리스케줄링 Upgrade 및 ASCS 통합 · 역간 및 센터연동시험 ○ 서브시스템 개발 및 시험

		· 라인모니터링 서브시스템 개발 · 수송관리 서브시스템 제작 · 라인경합 서브시스템 개발 ○ 외부테스터 개발 · 외부테스터 상세설계, 제작 · 외부테스터 자체 기능 시험 · 외부테스터 온라인 테스트모드 시험 ○ 홈 리스케줄링 연동시험 · 홈 리스케줄링 자체 기능 시험 · 인접역 연동시험 · 라인경합관리 연동시험(센터연동시험) ○ 종합연계시험센터 개발 및 시험 · 종합연계시험 계획 작성 · 현장 실 데이터 DB 구축 · CTC 에뮬레이터 서버와 타시스템 I/F 개발 · CTC 에뮬레이터서버 시험 및 기능보완 · 에뮬레이터 보완 제작 · CTC L/S 모니터링 콘솔 개발 · 종합연계시험센터 설치 · 종합연계 모의시험 · 결과분석 및 수정보완 ○ 현장시범시스템 구축 및 시험 · 현장시범시스템 구축 및 현장시험 · 시험결과분석 및 기능보완

여 백

2장 시스템 요구사항 분석

1절 국·내외 사례 조사

1. 국내 사례

국내 운행제어시스템은 열차집중제어 장치가 대표적이다. 통합 CTC 시스템은 각 지역사무소에 분산 배치되어 있는 사령 설비를 한 곳으로 통합하여 주요 선로구역의 열차 운행 관리와 열차 운행 시각, 유지보수 등을 집중 관리한다.

가. 통합 CTC 시스템 구성

통합 CTC 시스템은 중앙집중형 열차운행관리시스템이며, 사령설비와 현장설비로 구분된다. 사령 설비는 서버 시스템, 운영자 콘솔 및 부속 주변기기, 화상 시스템, 무정전 전원장치, 통신망장치(LAN) 등으로 구성되며, 현장설비는 역 정보 전송장치와 연동장치, 현장 신호 설비, 역 조작탁, 행선안내장치 등으로 이루어진다.

서버 시스템은 3개 지역별 CTC 서버 3조, 통신용 서버 2조, 스케줄 서버 및 개발용 서버 각 1조 등 결함 허용 시스템 구조의 컴퓨터 시스템과 Regulation 컴퓨터 3조 등이다. 34조의 운영자 콘솔은 선사령 콘솔 26조와 주사령 콘솔, 신호사령 콘솔 각 4조 등 이중계 PC 시스템과 프린터, 플로터 등 부속 주변기기 등이다.

사령 설비는 통신망장치의 터미널 서버를 포함한 이중계 통신망 장치와 접속되어 신뢰성을 높인 구성이다. 터미널 서버는 역 정보 전송장치와 통합 전송로를 연결하는 범용 인터페이스로 직접 접속된다. CTC 시스템은 본청 수송조정실 시스템, 고속철도 사령실 시스템, 고속철도 통합 정보 시스템, 철도 운영 정보 시스템, 전력 관리 시스템, 설비 관리 시스템 등 타 시스템과 통신망 장치를 통하여 정보를 교환할 수 있도록 구성된다.

(1) 물리적 구성

통합 CTC의 물리적 구조는 그림 2-1과 같이 CTC서버, 스케줄서버, 통신용서버, Regulation컴퓨터로 구성되어 진다.

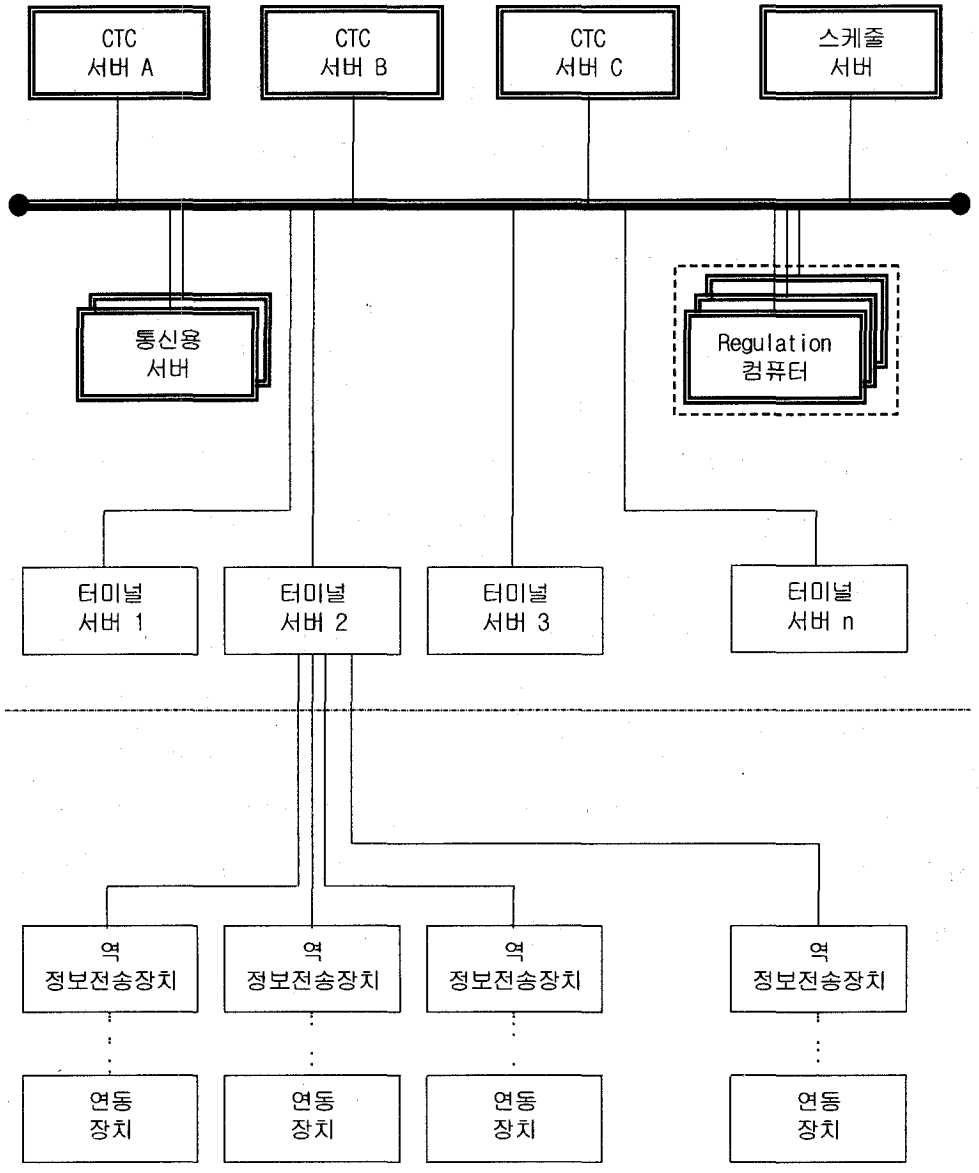


그림 2-1 통합 CTC 물리적 구성

(2) 논리적 구성

통합 CTC의 논리적 구성은 그림 2-2와 같이 현장으로부터 올라오는 정보는 터미널 서버에서 수신하고, 몇 대의 터미널 서버는 통신용 서버를 통해 CTC에 정보를 보낸다.

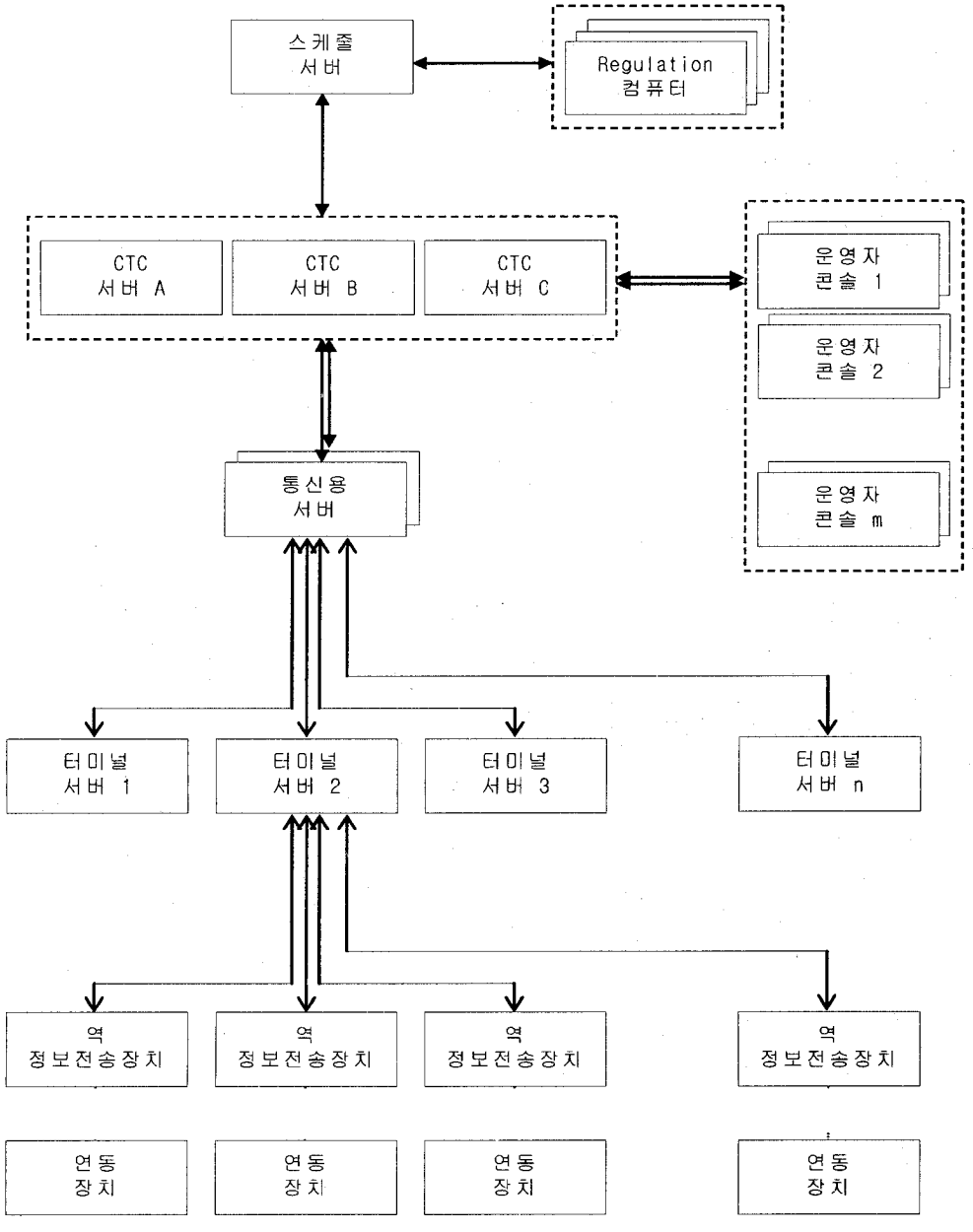


그림 2-2 통합 CTC 논리적 구성

나. 시스템 기능

(1) 시스템 관리 기능

통합 CTC 시스템의 모든 장치의 가동 상태를 감시하고, 그 이력을 관리하는 기능이다. 통합 CTC 시스템을 구성하는 컴퓨터들은 자신의 가동 상태를 포함한 '기기상태정보'를 주기적 또는 필요한 경우에 CTC 서버에 송신한다. 특히, 역 정보 전송장치는 자신의 상태와 연동장치의 가동 상태를 통신용 서버를 경유하여 '기기상태정보'의 형태로 CTC 서버에 전달한다.

(2) 각종 이벤트 관리 기능

통합 CTC 시스템의 운영 중에 발생하는 기기 장애 상태, 수동 진로 제어 및 선로전환기 전환 취급 등 운영자에 의한 수작업 실적, 표시 정보 및 제어 정보의 송수신 내역, 주요 설비의 상태 등에 대한 각종 이벤트를 관리하는 기능이다.

(3) 표시 및 제어 처리 기능

현장 신호 설비와 이와 관련된 역 설비에 대한 표시 처리와 제어 처리를 수행하는 기능이다. 표시 및 제어 처리는 연동 데이터 관리 기능과 운영 모드 관리 기능 등으로 세분된다.

(4) 열차 진로 제어 기능

열차 운행 스케줄에 의하여 자동으로 진로를 제어하는 기능, 그리고 운영자 콘솔에서 송신하는 수동 진로 제어 및 취소 등 '제어취급정보'에 의한 수동 진로 제어를 지원하는 기능이다.

(5) 열차 위치 추적 기능

현장에서 수신되는 '설비상태정보'를 이용하여 열차번호를 추적하는 기능이다. '설비상태정보'가 출발지 열차번호표시창에서 다음(목적지) 열차번호표시창으로 열차번호를 이동시키는 조건에 해당하는 '궤도회로점유정보'이면, 열차의 진행 방향

을 판단하여 열차번호를 목적지 열차번호표시창으로 이동시킨다. 운영자 콘솔에서 운영자의 입력에 의해 생성되는 열차번호의 변경, 삭제, 추가 및 이동 등 '열차번호 취급정보'를 수신하여 처리하는 기능을 수행한다. CTC 서버는 열차 위치 추적의 결과로 '행선안내정보'를 생성한다.

(6) 열차 운행 스케줄 관리 기능

열차의 운행 계획을 작성하고 운행 실적을 관리하는 기능이다. 열차 운행 실적 및 지연열차 보고서, 운휴/운전 지정 보고서, 수동 조작 이력/통계 보고서, 열차 운행 실적 보고서 작성 등의 운영 관리 기능이 있다. 운영자 콘솔에서 운영자의 입력에 의해 생성되는 열차 운행 계획의 변경, 취소, 조정 및 삽입 등의 운영자 수작업 처리를 지원하는 기능을 수행한다.

2. 국외 현황

국외의 열차운행제어시스템 현황으로 기존의 자동진호제어시스템과 동일본철도의 동경권수송관리시스템 ATOS (Autonomous decentralized Transport Operation control System)를 중심으로 조사하였다. 본 연구의 주된 목적중의 하나가 현재 우리나라에서 사용하고 있는 중앙집중 제어방식의 열차운행제어시스템의 단점을 개선할 수 있는 새로운 열차운행제어시스템을 연구하는 것이기 때문에 우리나라 간선철도와 운영환경이 비슷한 동일본철도의 열차운행제어시스템을 조사 분석한 것이다.

가. 기존의 열차운행제어시스템

(1) 자동진로제어시스템

일본에서는 1936년 처음 열차집중제어장치[1](CTC : Centralized Traffic Control) 시스템을 사용하였다. 이후 열차집중제어장치보다 향상된 기능이 필요하여 개발된 자동진로제어장치(PRC; Programmed Route Control)가 1967년 개발을 시작하여 1972년에 COMTRAC 시스템에 적용되었다.

1) 일본에서는 열차집중제어장치를 '집중원격 제어장치를 이용해서 열차지령이 일정 선구내의 열차운행상황을 직접 파악함과 동시에 신호기를 직접 제어하여 운전사에게 출발을 지시하는 운전방식'라고 정의한다.

일본에서의 열차운행관리 시스템의 계층구성은 크게 정보처리계와 진로제어계, 정보전송계로 구성된다. 최상위의 정보처리계(EDP: Electronic Data Processing)는 열차 운행다이아의 관리와 운행정리의 역할을 수행한다. 진로제어계인 자동진로제어장치는 열차추적과 진로를 제어하는 자동진로제어를 수행한다. 열차집중제어장치(CTC : Centralized Traffic Control)는 열차의 집중제어와 정보를 전송하는 시스템으로 구성된다. 그림2-3은 열차운행관리 시스템의 개념적 구성도를 나타내고 있다.

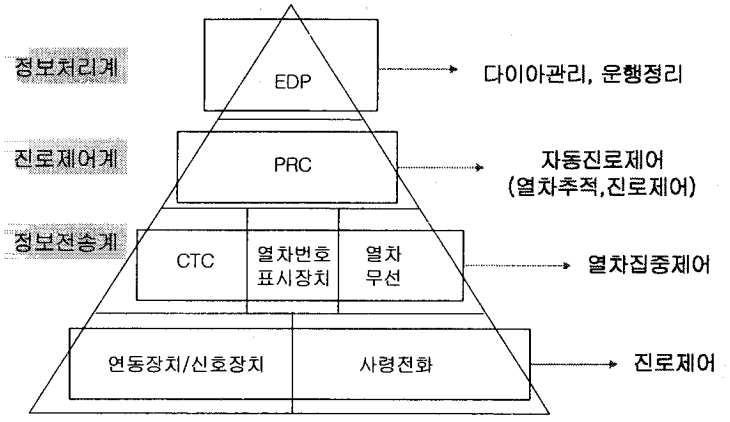


그림 2-3 열차운행관리시스템 계층

(2) 유연신호시스템 (Flexible Signalling System)

철도신호시스템은 대규모시스템으로 안전성 및 기능성에서 사회적으로 요구되는 기대를 만족시킬 필요가 있다. 신호시스템은 수명주기가 긴 반면에 수명주기 동안 기능의 개선이 어려울 뿐만 아니라 급격히 발전하는 정보기술에 뒤처지기 쉬워 유연성, 기능 변경성 및 외란대응성에서 해결해야 할 문제점을 아직 많이 가지고 있다고 판단된다. 이러한 문제를 해결할 수 있는 방법으로 일본전기 학회는 조사연구를 통하여 향후 개발되어야 할 유연시스템은 진화가 가능하고 정상이나 비정상 상태에서 유사한 기능을 수행할 수 있어야 한다는 시스템의 개념을 정리하였다. 이러한 개념을 만족하기 위해서는 자율분산방식이 필수적인 것으로 인식되고 있다.

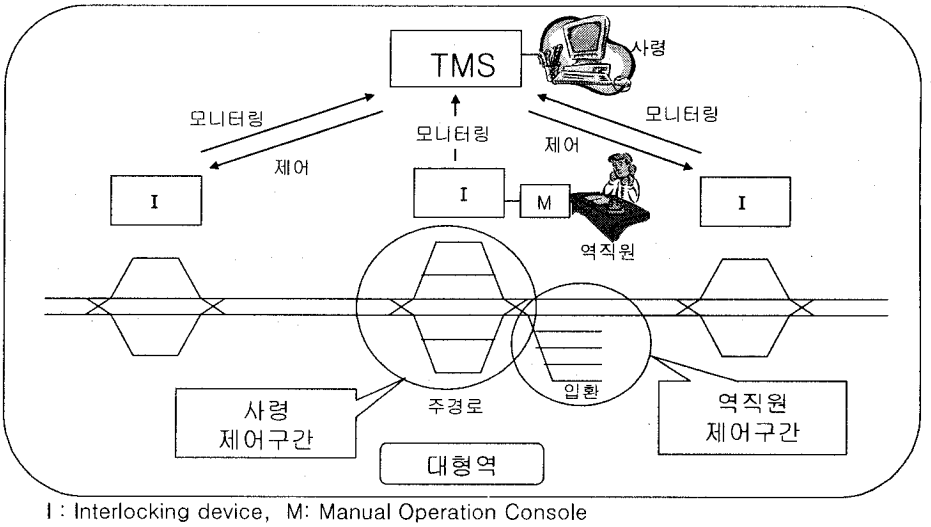
나. 동경권수송관리시스템

(1) 시스템 개발의 기술적 배경

일본 도쿄 도심지역 열차이용객의 수가 증가함에 따라 열차 운전시격은 2, 3분 간격으로 운전시격이 매우 짧게 운행된다. 이렇게 짧은 운전시격을 가진 열차운행 패턴을 사령자가 관리하여야 하는데 대부분의 업무는 전화로 정보를 수집하고 전송하였다. 열차운영상황이 정상일 때는 문제가 크게 발생하지 않는데 열차운행상황이 비정상일 때 여러 가지 문제들이 발생하기 시작하였다. 때문에 철도운행 제어에 대한 절차를 조직화하고 열차운행 제어를 자동화하기 위해 ATOS 시스템을 개발하였다. ATOS 시스템 개발자들은 문제점 분석을 위해 조사를 실시하였으며 이때 조사된 내용은 다음과 같다.

(가) 사령 · 역직원의 신호제어 수작업 처리 문제

대형역에서의 열차운행제어상의 가장 중요한 문제점은 그림 2-4에서 볼 수 있다. 와 같이 역을 제어하는 운영자가 2명으로 한사람은 사령실에 사령과 해당 역의 직원으로 이원화된 구조를 가지고 있다. 또한 사령은 주경로를 제어하고 역직원은 구내입환 구간을 제어한다. 그러나 이러한 이원화된 제어구조는 대형역에서 빈번히 발생하는 입환작업과 열차지연 등에 따라 역이 아닌 사령센터에 근무하는 사령이 그 업무를 수행하기가 불가능한 문제를 야기시켰다. 따라서 대형역의 신호들을 사령이 원격으로 자동제어를 하는 것은 어렵기 때문에 역에 근무하는 직원이 상시적으로 또는 일시적으로 로컬 수동제어를 하는 현상이 발생한다.



I : Interlocking device, M: Manual Operation Console

그림 2-4 ATOS 도입이전 대형역에서 열차운행제어 문제점

(나) 외란 해결의 어려움 문제

도쿄 도심지역의 열차운행은 열차운전 시격이 2분에서 3분으로 고밀도 운행구간이다. 이와 같이 운전시격이 짧은 구간에서 열차지연이나 사고와 같은 외란이 발생하면 지연 회복을 위한 사령의 결정은 오로지 사령의 경험적인 판단에 의해서 해결하였으며, 외란으로 변경된 다이아그램은 빠른 시간 내에 최적의 상태로 회복할 수 없는 문제를 가지고 있었다.

(다) 역에서의 여객정보 서비스의 부재

도쿄 도심지역의 신주쿠 역은 하루 이용객 평균이 약 740,000명이 이른다. 이와 같은 이용객이 많은 복잡한 역에서 열차 외란의 발생에 따라 열차가 지연이 발생할 때 역구내에 제공되는 여객안내 정보 서비스가 부족하였다. 또한 이러한 외란 상황에 대한 정보서비스도 역의 직원에 의하여 수작업으로 운영되었기 때문에 그 정확도와 신속성이 저하되는 문제점을 가지고 있었다.

(라) 선로 유지보수원의 안전 문제

일본 철도시스템에서는 신호제어 구간에서의 유지보수 작업의 승인과 유지보수 작업 개시는 역직원이나 제어센터에 근무하는 직원에게만 의존하여 운영되어 왔다. 시스템의 지원 없이 사람에 의해서만 운영되는 유지보수 작업 업무는 사고 예방과 안전을 보증하기에는 부족한 면이 있었다. 때문에 사령이나 역무원에 의한 유지보수 작업관리는 혼잡스럽고 유지보수에 대한 효과적인 작업시간이 줄어드는 단점을 가지고 있었다.

다. 시스템 구성

ATOS가 제어하는 시스템 구간은 그림 2-5와 같다. 17개 선구, 250개역, 총 1100km 트랙길이를 가진다. 그림에서 볼 수 있듯이 많은 역들이 노선과 노선간의 교차 역으로 구성되어 있다. 2002년 3월 기준으로 계획 구간의 68%가 완료된 상태이다. 그림 2-6는 ATOS시스템의 구성도이다.

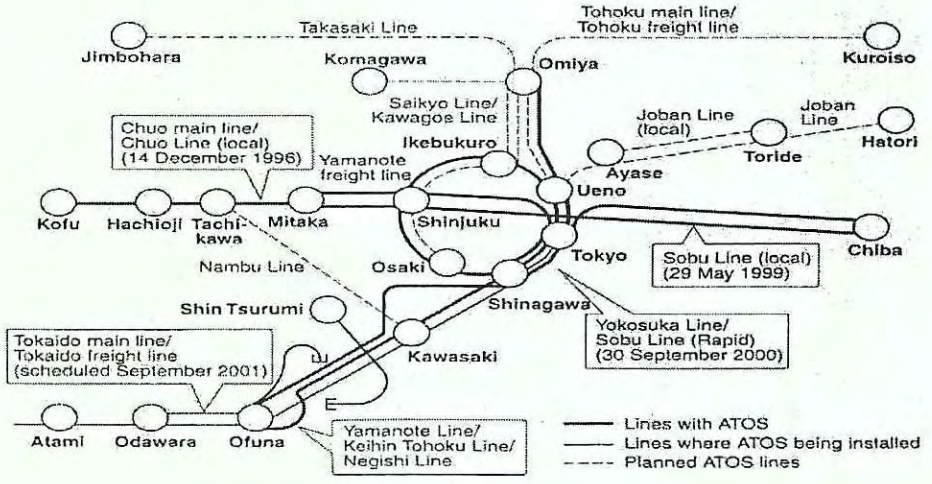


그림 2-5 도쿄지역의 ATOS 제어 구간

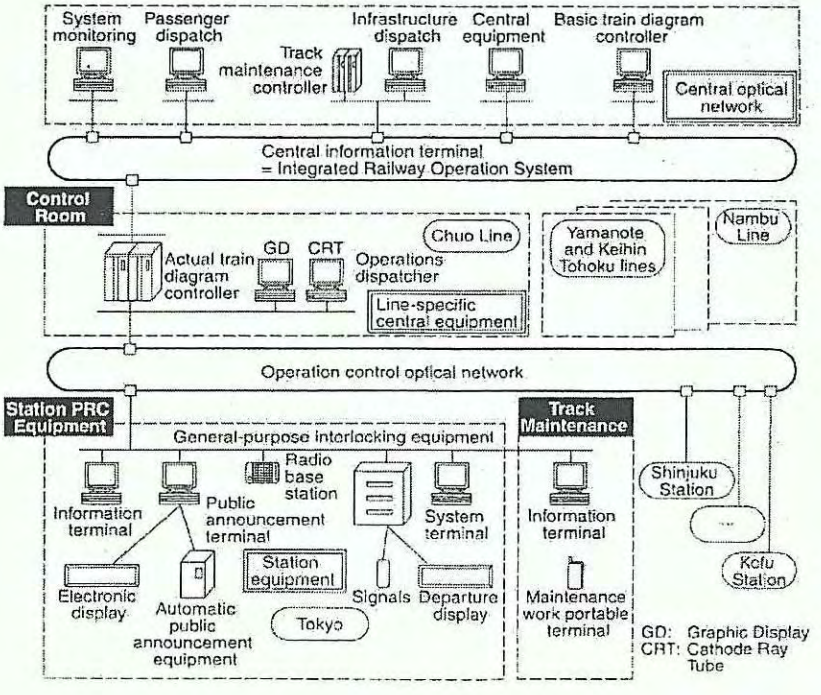


그림 2-6 ATOS 시스템 구성도

라. ATOS의 효과

ATOS는 1993년 역에서 시스템 도입을 시작한 이후로 현재까지도 지속적인 확장을 해 가고 있다. 지금 Chuo 라인과 같은 전체 선구에서 1996년 12월에 운영을 시작했으며 Yamanote 라인과 Keihintohoku 라인에서 1998년에 운영을 시작했고, 1999년 3월에 Ochanomizu와 Chiba사이에서 운영을 시작했다.

열차 운행 제어에 대한 커다란 운영 환경의 변화를 꽤한 ATOS는 도쿄 도심 지역의 초고밀도 선구에서 수행되었고 열차의 운영 환경은 일반 수작업으로 운영되는 것 보다 교통 외란이 있는 동안에 원활히 수행되었다.

단계적인 시스템 구축방법을 적용한 시스템은 항상 발전한다. 예를 들면 Yamanote와 Keihintohoku 선구에 사용된 2호 선구의 구축을 통해서 3년 동안 총 173회에 역시스템 설치가 수행되었으며 중앙시스템 설치는 14차례 수행되었다. 이러한 시스템 설치운영은 하룻밤 동안의 매우 짧은 시간동안에 수행되었다. 이것은 시스템 변경에 따른 중요 문제가 없이 수행되었다. 이러한 측면으로 자율분산 시스템 기술은 열차 안전과 안정적인 운영에 매우 큰 기여를 하였다. ATOS 구현에 따른 주요 효과를 정리하면 표 2-1과 같다.

표 2-1 ATOS 구현에 따른 주요효과

효과	주요 내용
복잡하고 고밀도 선구에서의 열차운행관리의 실현	▪ 모든 역의 중앙사령 제어기능 ▪ 경합되는 열차들에 대한 우선순위 판단이 사령의 개입 없이 완전 자동화로 실행 ▪ 외란 발생시 진보된 맨·머신 시스템을 사령이 활용하여 빠른 다이아 회복이 가능 ▪ 열차간의 간격제어를 할 수 있는 열차그룹제어 기능
효율적인 철도관리시스템의 실현	▪ 대형역에서 주선로 열차들의 자동경로 제어 ▪ 유지보수 구간의 직원이 궤도 유지보수 작업을 단독으로 실행할 수 있는 것을 허용
선로 유지보수자의 안전 실현	▪ 연동제어 시스템은 열차와 궤도유지보수 작업간의 사고를 방지함
진보된 여객정보시스템의 실현	▪ 외란 발생 정보의 지속적 서비스 제공 ▪ 모니터-된 열차 교통정보는 역직원에게 중계됨 ▪ 사고 정보는 중앙 사령실에서 역 안내장치로 동시에 표시됨
효율적인 유지보수시스템의 실현	▪ 원격진단 및 유지보수를 개방형타입의 OS와 표준 통신 프로토콜을 사용하여 실행할 수 있음

3. 국·내외 기술현황 분석 결과

국내의 열차운행제어시스템은 중앙집중 제어방식을 채택하고 있는 반면에 동일본철도는 자율분산제어방식을 기본으로 하고 있다. 따라서 우리나라의 경우 철도공사에서 운영하고 있는 모든 간선철도를 한 곳에서 제어하고 있다. 단, 현재 경부고속철도는 별도로 분리되어 운영되고 있다. 그러나 동일본철도의 경우 중앙에서 열차운행스케줄을 관리하고 있으나 실제 제어형태는 노선별 제어형식을 채택하고 있다.

각 방식의 기능적 특징, 시스템 구성방식, 구축방식 및 확장성 등 특징은 표 2-2와 같다.

표 2-2 CTC와 ATOS의 주요 차이점 비교

시스템 / 항목	CTC	ATOS
기 능	- 표시·제어, 데이터관리 - 이벤트 로깅, 프로세스 감시 - 외부인터페이스	- 열차스케줄 관리·수정 - 역의 경로제어, 유지보수 작업관리 - 역에서 고객정보
시스템 구성방식	중앙집중형 제어방식 (지역별 제어방식)	자율분산형 제어방식 (노선별, 역별 제어방식)
구축방식	일괄 구축 방식	단계적 구축방식
통신 방식	중앙시스템을 목적지(Destination)로 한 마스터-슬레이브 방식	컨텐츠코드를 가진 브로드캐스팅 통신 방식
확장성	- 역시스템의 상황에 따라 중앙시스템의 변경이 필요 - 온라인 확장이 용이치 않음 - 새 S/W버전 출시시 모든 시스템의 교체가 요구됨	- 중앙시스템과 역시스템은 단계적 구축을 할 수 있음 - 기능과 성능의 온라인 확장 가능 - S/W 버전 관리

국·내외 기술개발 현황 분석은 크게 두 가지 관점에서 접근할 수 있었다. 하나는 철도의 열차운행관리 시스템을 위한 철도만의 운영제어 기술이다. 또 다른 하나는 진보적인 열차운행관리 시스템을 위한 적용기술이다.

첫 번째, 열차운행관리 시스템에서의 운영제어 기술이다. 국내기술은 통합CTC 구축을 위한 사업 현황에서도 알 수 있듯이 통합 CTC시스템의 목표는 시스템 운

영의 집중화이다. 반면에 일본의 시스템은 역의 자율성을 두고 시스템 분산화로 기술이 개발되어 가고 있다. 시스템의 질적인 측면에서 고려해 보면 아직 일본 시스템의 자동제어 부분에서 국내 시스템은 미진한 기능을 가지고 있다. 이는 일본의 ATOS 시스템 개발 초기의 환경에서 알 수 있듯이 국철에서 사철로 바뀌는 환경에 적응하기 위해 인력절감의 측면에서의 자동화 시스템 도입의 박차를 가한 것도 하나의 이유가 되겠다.

국내 철도운영 환경을 고려해 보면 우리나라 대형역도 일본의 경우와 마찬가지로 열차 시격이 짧고 열차 입환 작업등으로 중앙시스템의 제어는 현실적으로 불가능한 상태이다. 경부선만 살펴보더라도 4급 역은 8개역이며 이중 7개 역이 상시 로컬 상태로 운영되고 있다. 때문에 상시로컬로 운영되고 있는 역 담당자는 과중한 업무부담을 안게 된다. 때문에 현재 비효율적으로 운영되고 있는 역의 운영효율 증대를 위해서 역시스템의 구축이 필요하다. 이 시스템은 역에서의 선로용량 증대뿐만 아니라 역 자동운행에 대한 국내 시스템의 새로운 비전을 제시해 줄 것으로 기대된다.

두 번째, 열차운행관리 시스템 구축을 위해 적용된 세부기술이다. 정보기술 측면에서만 보면 국내기술은 여러 선진국과 비교해 동등한 수준을 가지고 있으며 그 이상의 실력도 가지고 있다고 할 수 있다. 그러나 열차운행관리 시스템이 이 시스템 자체로서의 역할을 충분히 활용하기 위해서는 타 시스템과의 원활한 인터페이스가 이루어져야 한다. 일본의 경우에서는 시스템의 신뢰성을 위해 결함허용시스템 구축을 위해 자율분산시스템을 철도에 적용하여 성공적으로 시스템을 운영하고 있다. 이는 철도운행관리 시스템이 항상 온라인 운영이 필요하므로 온라인 확장에 대한 요구조건을 만족할 수 있으며 각 역시스템이 독립적으로 수행되기 때문에 시스템의 구축 측면에서도 단계적 구축방식을 적용하여 지속적으로 변화하는 내부 시스템 사용 고객의 요구를 충분히 반영하여 시스템을 개발할 수 있을 뿐만 아니라 구축비용 측면에서도 상당한 비용절감 효과를 가질 수 있다.

국내의 시스템 개발환경을 살펴보면 이러한 대규모 지역을 커버하는 열차운행관리시스템의 업그레이드나 기능향상을 위해서 장기간에 많은 비용을 투자해서 개발을 수행하여 왔다. 국내에서도 처음 자율분산 개념을 도입한 역시스템 개발이 완료되면 이러한 부분은 상당히 해소 될 수 있을 것으로 기대된다.

2절 상시 로컬역 현황 조사 분석

현재 열차집중제어장치의 운영모드는 크게 중앙제어센터에서의 원격제어모드와 역 조작반(LCP)에 제어권한이 부여되는 로컬모드로 나뉘어져 있다. 열차집중제어장치의 본래 목적인 전체노선의 효율적 중앙관리를 위함이지만 대부분의 대형역들은 관제센터에서 원격으로 제어를 하지 않고 로컬모드 운영하고 있기 때문에 효율이 저하되고 있다. 때문에 이러한 상시로컬역의 현황분석을 통해 본 연구과제에서 수행하려고 하는 역시스템 개발을 위한 요구사항 및 개선사항을 도출하기 위해 상시로컬상태로 운영되는 여객 열차의 운행이 빈번한 역과 화물열차 운영이 빈번한 두 역을 조사하였다. 조사결과의 정리를 위해 상시로컬 운영이유, 상시로컬역의 문제점 분석, 상시로컬 역 운영자들의 요구사항을 분석 정리하였다.

1. 상시로컬 운영 현황

가. 여객중심의 상시로컬역 현황분석

여객중심의 대표적인 상시로컬역으로 청량리역을 들 수 있다. 청량리역은 그림 2-7과 같이 4개의 선이 겹치도록 구성되어 평상시 역무원의 업무량이 많고, 열차흐름의 병목현상이 잦은 역이다.

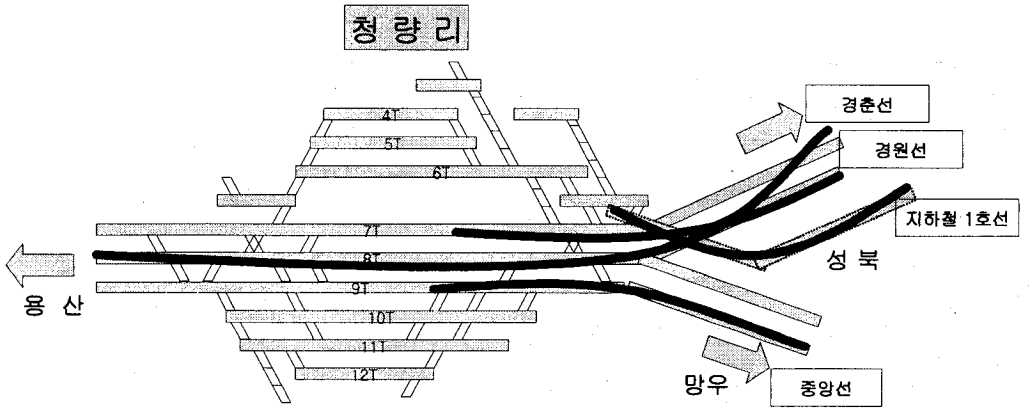


그림 2-7 여객 중심의 상시로컬역

(1) 조직 구성

청량리역의 조직 구성은 다음 그림 2-8과 같이 열차운용팀, 신호취급실, 수송팀,

검수팀으로 구성된다. 이 중 열차운용은 열차운용팀과 수송팀에서 주로 담당한다. 열차운용팀은 열차운용팀장이 거의 대부분의 업무를 하고 있고 그림2-8과 같이 역 조직 내부 뿐만 아니라 외부 조직과도 업무가 연계되는 곳이 많다. 수송팀은 청량리역의 넓고 복잡한 구조 때문에 3개조로 나뉘어 각각의 업무를 분리하고 있다.

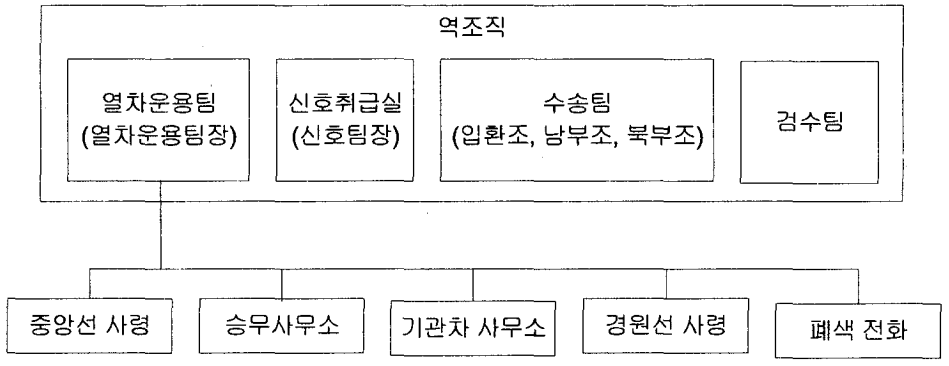


그림 2-8 청량리역 조직 구성

(2) 운용 현황

(가) 열차운용팀

① 열차의 착발관련 업무 처리

- o 실시간 열차위치 확인(CRT 및 열차체류표시반 활용)
- o 열차 운행 상황 파악(KROIS 활용)
- o 출발열차 출발선 지정, 도착열차 도착선 지정
- o 열차별 차입선 조성 독려
- o 관내 열차 운행 상황 파악 및 장표 정리 취급
- o 이례상황 발생시 보고 및 관계부처 통보
- o 승하차 여객 유도 안내

② 여객 열차 조성

③ 역 구내 진로 설정 및 신호 취급

④ 인접 역과의 작업 협의

(나) 신호취급실

열차운용팀과 신호 취급실이 떨어져 있어 신호팀장은 열차체류 표시반을 보고 열차 운용팀장의 진로 승인이 나면 신호처리를 한다.

(다) 수송팀(3인 4개조, 하루 20명 근무)

- ㅇ 입환조 . 열차를 조성하여 본선에 갖다 놓는 작업
- ㅇ 남부조 : 플랫폼 도착 열차를 측선으로 전선 하거나 정비실로 옮기는 작업
- ㅇ 북부조 : 플랫폼에 갖다놓은 열차에 기관차를 붙이는 작업

(라) 검수팀

나. 화물 중심의 상시로컬역 현황분석

망우역은 그림 2-9에서 볼 수 있듯이 제천조차장에서 조성되어 오는 화물 열차를 받아 서울 지방의 모든 화물을 담당하는 화물 중심의 상시로컬역이다.

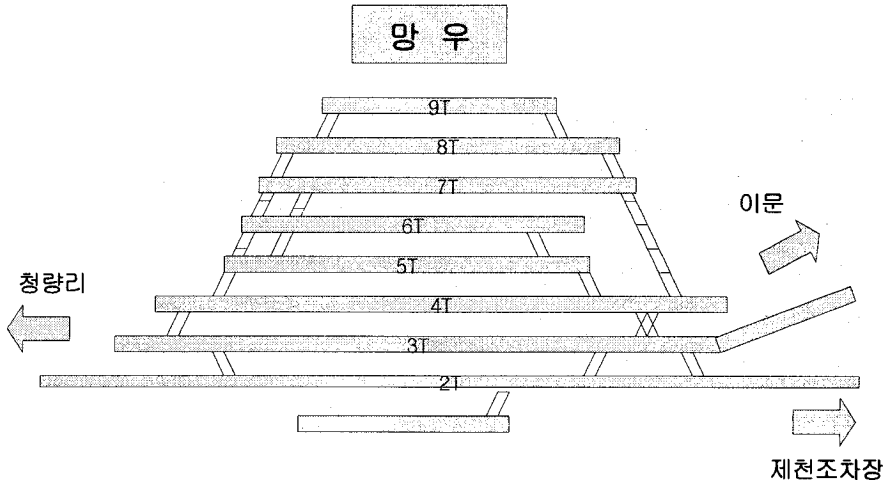


그림 2-9 화물중심의 상시로컬역

(1) 조직 구성

망우역의 조직 구성은 그림2-10과 같이 열차운용팀, 신호취급실, 수송팀으로 구성된다. 이 중 열차운용은 열차운용팀과 수송팀에서 주로 담당한다. 열차운용팀은 열차운용팀장이 거의 대부분의 업무를 하고 있고 그림2-10과 같이 역 조직 내부뿐만 아니라 외부 조직과도 업무가 연계되는 곳이 많다. 수송팀은 망우역 특성에 맞추어 내근 수송원과 외근 수송원으로 구분되어 내근 수송원은 입환스케쥴 작성, 입환기 스케쥴작성, 입환지시서 발행 등의 업무를 하고 외근 수송원은 입환지시서를 기관사에게 발행하고 입환 작업을 한다.

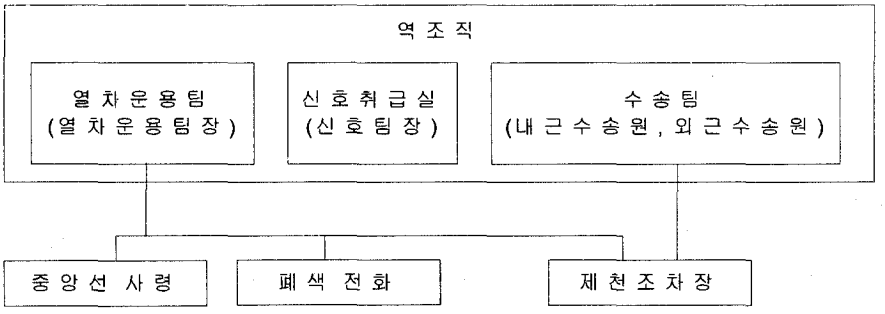


그림 2-10 망우역 조직 구성

(2) 운용현황

(가) 열차운용팀

① 열차의 착발관련 업무 처리

- 실시간 열차위치 확인(CRT 및 열차체류표시반 활용)
- 열차 운행 상황 파악(KROIS 활용)
- 출발열차 출발선 지정, 도착열차 도착선 지정
- 열차별 차입선 조성 독려
- 관내 열차 운행 상황 파악 및 장표 정리 취급
- 이례상황 발생시 보고 및 관계부처 통보

② 화물 열차 조성

③ 역 구내 진로 설정 및 신호 취급

④ 인접 역과의 작업 협의

(나) 신호취급실

열차운용팀장과 신호취급실이 붙어 있어 열차 운용팀장이 진로 승인을 하면 바로 신호처리를 한다.

(다) 수송실

① 내근 수송원

- ㅇ 입환선 사용계획 작성
- ㅇ 입환기 스케줄 작성

② 외근 수송원

- ㅇ 기관사에 입환지시서 전달
- ㅇ 입환작업

다. 상시로컬 운영이유

- ㅇ 본선에 지장을 주는 입환 업무가 있는 역은 중앙에서 제어할 수 없기 때문에 로컬 취급함
- ㅇ 스케줄 변경시 플랫폼에서 경합상황을 중앙에서 해결하지 못함
- ㅇ 통합관제실에 표시되지 않는 선로가 현장역에 많이 있음

라. 상시로컬 역 문제점 분석

- ㅇ 복잡한 선로 특성으로 인하여 원격제어모드 사용이 불가능하여 로컬 모드로 운영
- ㅇ 로컬모드 사용으로 인항 중앙 시스템을 활용한 직접제어 불가능
- ㅇ 열차운용팀장(원)의 know-how에 의한 운영
- ㅇ 열차운용팀장(원)의 업무가 복잡하고 수동으로 처리됨
- ㅇ 열차번호 확인하는 업무와 열차 도착시간을 확인하는 업무비중이 크다.

- 궤도회로가 설치된 구간 이외의 상황은 열차운용팀장이 즉시 파악하기 곤란
- 도착/출발 열차의 도착선 지정시에 열차 경합이 발생할 경우 열차운용팀장의 판단에 의존하여 결정함
  - 경합처리를 위한 시간 및 정보의 양이 부족함
  - 열차운용팀장의 판단에 의존함으로써 사고 발생 가능
  - 운영팀장이 업무처리를 위한 결정시 고려해할 정보가 많다.
- 운영자가 역 업무와 관련된 불필요한 많은 정보를 관리한다.
- 사전 선사용 계획 없는 열차운용팀장에 의한 선관리
- 조상, 조하와 같은 관제사에 의한 운전명령 발생시 로컬에서 취급되던 업무가 복잡해짐
- KROIS(Korea Rail Operation Information System)를 통한 비실시간 정보를 제공받고, CTC를 통한 실시간 정보를 동시에 수신하여 처리함으로써 업무처리가 난해함

마. 상시로컬 역 운영자들의 요구사항 분석

- 상시 로컬 운영을 위한 시스템 구축 필요
- 열차번호/열차도착시간을 확인하여 입력하는 시스템 필요
- 부본선 구간 궤도 회로 설치 필요
- 경합처리를 위한 경합처리 시스템 필요
- 선 관리를 위한 선사용 계획 작성 시스템 필요
- 조상, 조하와 같은 예외 상황 처리를 위한 시스템 필요
- 역에 필요한 정보를 제공받아 처리하기위한 시스템 필요
- 입환 관리를 위한 시스템 구축 필요
- 업무협조를 위한 시스템 필요
- 예비선 활용계획 작성, 입환기 스케줄 작성을 위한 시스템 필요
- 고장이나 열차의 증감편성과 같은 예외상황 처리를 위한 시스템 필요

○ 예비선 관리 시스템 필요

○ 입환 작업 수행을 위한 지원 시스템 필요

○ 화물조성 효과를 증대시키기 위한 시스템 필요

○ 부서간 업무 협조를 위한 시스템 구축 필요

○ 열차운용팀장의 업무를 지원하기위한 시스템 필요

○ 부서간 협력 업무를 지원하기 위한 정보 시스템 필요

○ 부본선을 효율적으로 활용하기위한 시스템 필요

○ 선로의 효율적 운영을 위한 시스템 필요

3절 시스템 요구사항 분석

1. 현장조사에 따른 요구사항

가. 열차운용 담당자 요구사항

○ 출발·도착열차의 출발선 도착선 지정업무의 효율향상 요구

○ 역구내의 실시간 열차위치 확인범위의 확대 요구

○ 외란 해결을 위해 사용자가 원하는 정보의 빠른 획득 요구

○ 역구내 타부서의 효율적인 정보교환 방법의 요구

○ 열차 운행제어 업무 이외의 업무 경감 요구

○ 역 선로 부족으로 인한 선로 확충의 요구

○ 정보지원 시스템의 신뢰성 확보 요구

나. 신호취급 담당 책임자 요구사항

○ 시스템의 신뢰성 확보 요구

다. 수송취급 담당 책임자 요구사항

○ 화물취급역과의 정보교환 요구

○ 입환 작업을 위한 선로 부족으로 인한 선로확충 요구

라. 사용자 요구사항 분석

사용자 요구사항 분석은 상시로컬역 현장조사를 통해 얻을 결과물을 반영하였다.

(1) 열차운영 업무 효율화

수작업으로 이루어지고 있는 역에서의 경로제어 업무 자동화 및 착발열차의 선로지정, 착발승인 업무의 효율화를 위함

(2) 역구내 선로 부족 해결

제한된 선로자원의 효율화를 위해 최적화기법을 적용한 선로사용을 지원하기 위함

(3) 타부서와의 업무 효율화

역에서 발생하는 각종 업무(열차운용, 입환, 고객정보서비스 등) 담당 부서와의 원활한 의사교류 등을 시스템적으로 지원하기 위함

(4) 입환 업무 효율화

여객 및 화물열차의 입환 업무 효율화를 위해 시스템적으로 지원하기 위함

(5) 타 역과의 정보 교류

업무상 관련된 역(예를 들면, 화물차량의 조성과 관련된 정보를 얻을 수 있는 타 역)에서의 필요정보의 수집 및 인접역에서의 요구 데이터의 교환

2. 자율분산형 시스템 요구사항

분석된 시스템 요구사항을 만족하기 위하여서는 자율분산방식의 시스템이 요구된다. 일반적으로 알려져 있는 자율분산형으로 시스템을 구축하기 위하여 필요한 요구사항은 다음과 같다.

가. 자율분산형 서브시스템 요구사항

(1) 자율분산 논리관점에서의 요구사항

- 자율분산형 서브시스템들이 모여서 전체시스템이 구성되어야 한다.

- 자율분산형 서브시스템은 역 특성에 맞는 목적과 기능을 가져야 한다.
- 전체시스템 관점에서 하나의 서브시스템 기능 불가동은 정상이어야 한다.

(2) 자율제어성에 대한 요구사항

- 하나의 자율분산형 서브시스템이 기능불가동이어도 타 역의 동일 목적의 자율분산형 서브시스템은 스스로를 제어하여야 한다.
- 하나의 자율분산형 서브시스템이 자율성을 위해 센터에서 생성한 정적데이터인 열차스케줄 정보를 서브시스템 내에 저장하여야 한다.

(3) 균질한 구조에 대한 요구사항

- 자율분산형 서브시스템은 입출력 관계가 동일한 구조를 가져야 한다.
- 어떠한 자율분산형 서브시스템도 접속할 수 있어야 한다.
- 각각의 자율분산형 서브시스템은 여러 프로세스를 가질 수 있어야 한다.

(4) 평등한 기능에 대한 요구사항

- 자율분산형 서브시스템은 스스로 판단하고 제어할 수 있어야 한다.
- 자율분산형 서브시스템은 스스로의 판단으로 협조할 수 있어야 한다.
- 자율분산형 서브시스템은 타 서브시스템으로 명령을 내리지 않아야 한다.
- 자율분산형 서브시스템은 타 서브시스템으로 지령을 받아 수행하지 않아야 한다.

나. 데이터필드 시스템 아키텍처

(1) 일반 데이터필드 아키텍처

자율분산형 서브시스템의 구조적 균질, 기능적 평등, 정보적 국소 조건을 함께 합친 시스템 아키텍처를 데이터필드 아키텍처라 부른다. 이러한 데이터필드 아키텍처에서는 자율성을 갖는 서브시스템 즉, 자율분산형 서브시스템은 단지 데이터필드만을 통해서 다른 서브시스템과 협력한다. 이러한 방식의 중요 차이점은 데이터를 관리하는 서브시스템을 설치하지 않고 데이터가 흐르고 있는 것으로 보고 서브시스템이 자율적으로 데이터를 선택하여 이용하고 처리될 수 있도록 한 것이 데이터필드 아키텍처이고 데이터필드란 데이터가 흐르는 곳이다. 이러한 데이터필드 아키텍

처의 자율성을 이루기 위한 조건과 시스템 구조는 아래 그림2-11과 같다.

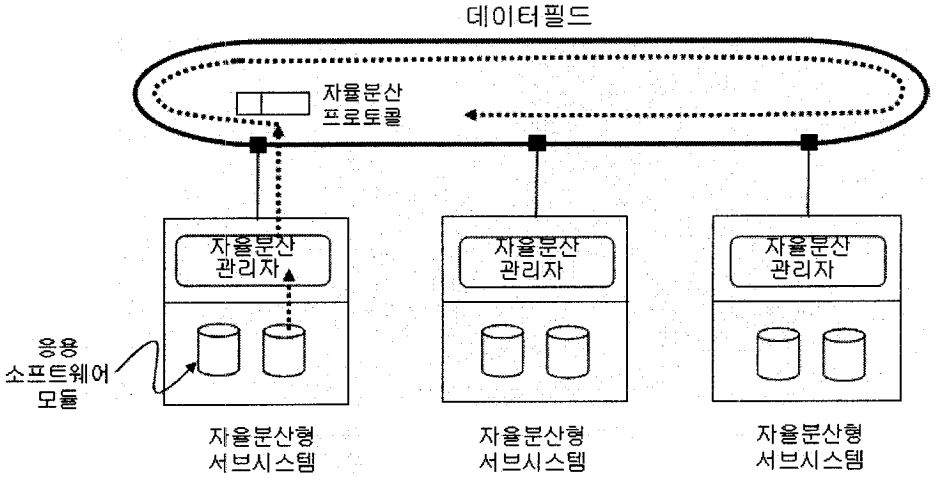


그림 2-11 데이터필드 아키텍처의 논리적 구성

(2) 협조 데이터필드 아키텍처

협조데이터필드 아키텍처에서는 데이터필드 아키텍처에 다음과 같은 구조들을 사용된다.

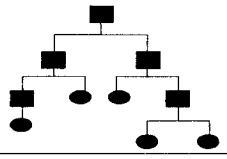
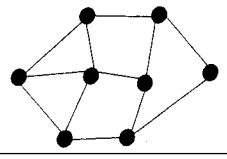
- 데이터필드 번호 : 일반구조에서는 비록 데이터필드가 시스템에 단지 한 개만 존재하지만 협조 데이터필드 구조에서는 시스템 내에 여러 개의 데이터필드를 가질 수 있다. 데이터필드 번호는 시스템내의 데이터필드를 구별하기 위함이며 데이터필드 번호는 각각의 도메인 필드에 따라 유일하게 할당된다.
- 도메인필드 번호 : 데이터필드의 상위의 개념으로 도메인필드를 정의한다. 도메인 필드에서는 여러 개의 데이터필드가 있으며 도메인필드 번호는 시스템내의 도메인 필드를 구별하기 위함이다. 도메인필드 번호는 전체시스템에 유일하게 할당된다.
- 논리노드 번호 : 일반구조에서는 컴퓨터 노드를 서브시스템(또는 atom)이라 부른다. 컴퓨터 노드는 논리노드라 부르며 논리노드번호는 데이터필드에서 논리노드를 구별하기 위함이다. 결과적으로 논리노드 번호는 각각의 데이터필드에 유일하게 할당된다.
- 내용코드 : 협조데이터필드 아키텍처 구조에서는 모든 데이터가"내용코드"를 가지고 있으며 각각의 데이터필드 내의 데이터의 내용 측면에서 단일하게 정

의된다. 그래서 다른 데이터필드 내의 동일한 내용코드는 시스템 내에 다른 하나이며 다른 도메인과 다르다. 일반구조에 있는 일관성과 상호운영성을 유지할 수 있다 왜냐하면 일반 데이터필드내에서 새로운 구조내의 데이터필드의 하나로 정의할 수 있기 때문이다.

다. 분산시스템의 비교

자율분산개념을 계층형 분산개념과 비교하면 표 2-3과 같다. 본 연구의 시스템 요구사항을 만족하기 위해서는 자율분산방식이 더 유리함을 알 수 있다.

표 2-3 계층형 분산시스템과 자율분산 시스템과의 비교

		계층형 분산시스템	자율분산 시스템
시스템구조			
		상·하위의 구조를 가지므로 각 서브시스템간은 지배관계가 있다	마스터-슬레이브 관계가 아닌 각자의 서브시스템끼리 상호협조적으로 행동한다.
장단점	고장시	상위의 서브시스템이 고장이면, 그 하위의 서브시스템 전체의 기능저하를 유발시킨다.	하나의 서브시스템이 고장일지라도 타 서브시스템은 각각의 담당영역을 임의로 제어가능하기 때문에 시스템 전체에는 지장이 없다.
	제어성	중앙시스템에 정보가 집중되어 있으며 제어정도가 양호하다.	필요시 각각의 서브시스템의 상태를 추정할 필요가 있다.
	통신 부하	중앙시스템의 통신 부하가 가중될 경우가 있다.	이기종 시스템간의 정보데이터 교환시 통신 부하가 가중될 경우가 있다.
	시스템 확장성	시스템 확장을 위한 작업량이 많아진다.	서브시스템만을 고려하므로 확장이 용이하다.
	비용측면	전체시스템을 고려하므로 비용이 많이 소요된다.	서브시스템 구축후 단계적으로 추가하기 때문에 적은 비용이 소요된다.

여 백

3장 대형·고밀도운행 철도역 자동운행제어 시스템

1절 자율분산 열차운행관리 시스템 구성

대형·고밀도운행 철도역 자동운행제어시스템 (Autonomous Decentralized Traffic Management System, 이하 ADTMS) 개발을 위해 역을 구성하는 역 시스템과 라인단위의 센터역할을 하는 노선관리 시스템을 개발하였고 전체 시스템의 구성은 자율분산 열차운행관리 시스템이라 한다. 아래 그림 3-1은 자율분산 열차관리 시스템의 각 서브시스템 구성도를 나타낸 것이다.

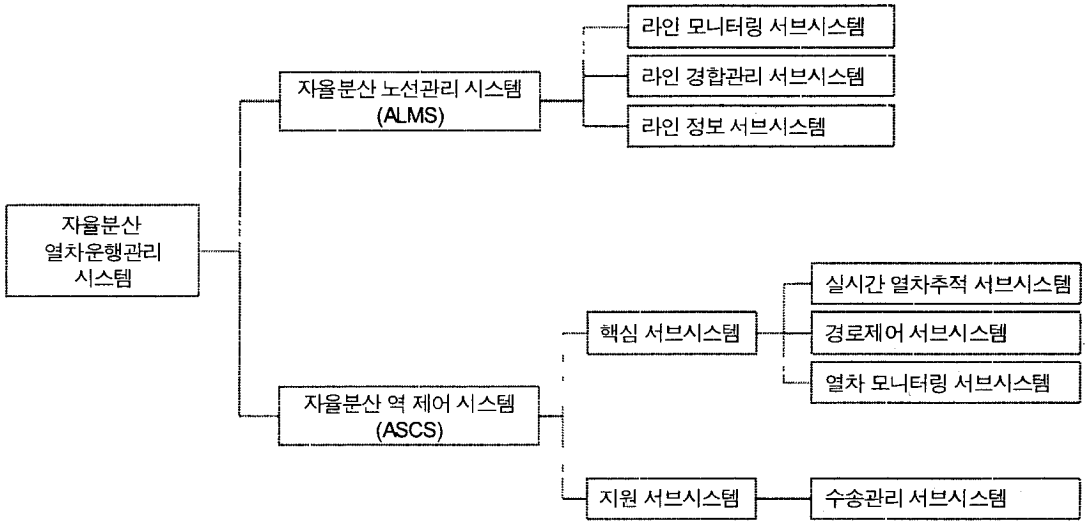


그림 3-1 자율분산 열차운행관리 시스템 구성도

자율분산 노선관리시스템(ALMS; Autonomous Line Management System)은 라인단위의 열차운행상황과 역 현황을 모니터링 하는 라인모니터링 서브시스템, 라인단위의 열차경합과 해소를 하는 라인경합관리 서브시스템, 라인단위의 스케줄을 조정하는 라인정보서브시스템으로 구성된다.

역 시스템은 자율분산 역제어시스템(ASCS; Autonomous Station Control System)으로 구성되며 열차위치 추적과 열차번로를 관리하는 실시간 열차추적서브시스템과 열차의 진로를 제어하는 경로제어 서브시스템, 역구내의 운행상황을 모니터링 하고 수동제어를 수행할 수 있는 열차모니터링 서브시스템, 열차의 입환을 관리하는 수송관리 서브시스템으로 구성된다. 그림 3-2은 라인센터와 역 시스템의 시

스템 구성도를 나타낸 것이다.

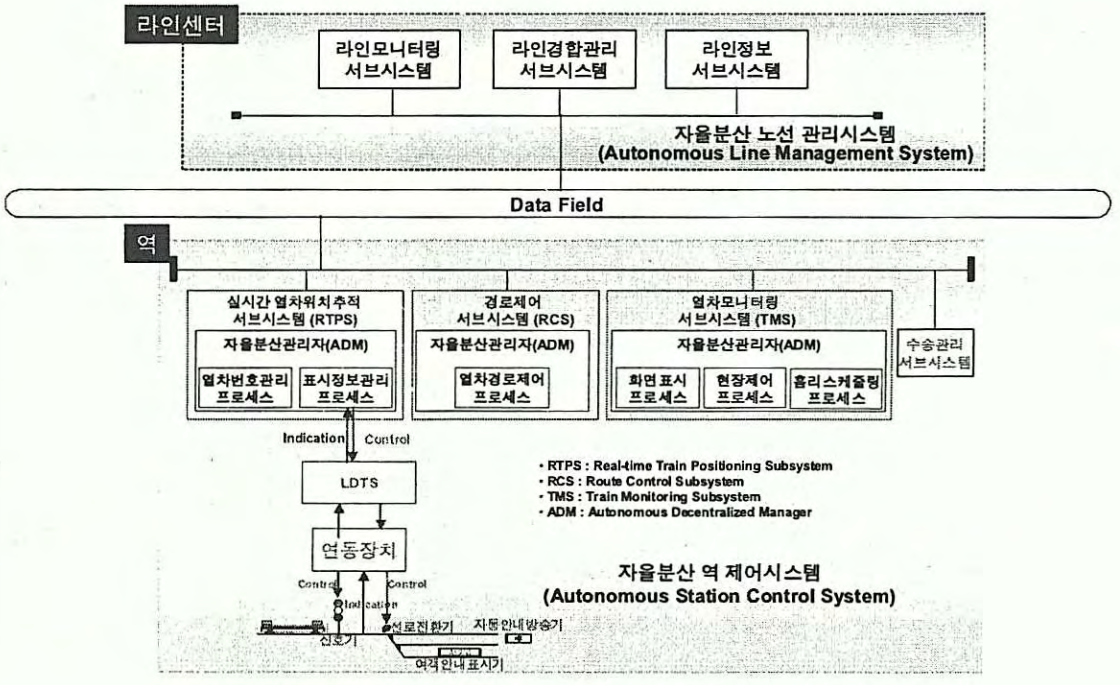


그림 3-2 자율분산 열차운행관리 하드웨어 구성도

2절 자율분산 역제어시스템

자율분산 역제어시스템은 대형·고밀도운행 철도역 자동운행제어시스템(ADTMS)의 역시스템에 해당한다. 자율분산 역제어시스템은 실시간 열차위치추적 서브시스템, 경로제어 서브시스템, 열차모니터링 서브시스템으로 구성한다. 또한 대형·고밀도 철도역에서의 열차 입환 작업을 지원하는 수송관리 서브시스템이 있으며 모든 서브시스템을 자율성과 협조성 지원과 통신 부분을 지원하는 자율분산관리자(ADM; Autonomous Decentralization Manager) 시스템으로 구성된다.

1. 실시간 열차위치추적 서브시스템

핵심 서브시스템의 하나인 실시간 열차위치추적 서브시스템(RTPS; Real-time Train Positioning Subsystem)은 역내에서 운행하는 모든 열차들에 대한 열차의 위치 추적과 열차번호를 관리하기 위한 기능을 가지며, 역 정보 전송장치(LDTS)로부터 수신되는 선로변 장치들(신호기, 선로전환기, 궤도회로 등)의 상태를 기록한

다.

실시간 열차위치추적 서브시스템은 기능의 특성에 따라 열차번호관리 프로세스, 표시정보관리 프로세스로 구성된다.

가. 열차번호관리 프로세스

열차번호관리 프로세스는 표시정보관리 프로세스로부터 수신되는 선로변 장치들에 대한 표시정보와 경로제어 서브시스템의 열차경로제어 프로세스로부터 수신되는 열차스케줄 정보를 기초로 하여 정의된 열차추적 논리에 의해 열차 위치를 추적한다. 역 구내로 진입하는 열차는 열차번호 인식장치에 의해 해당 열차의 실 열차번호를 인지할 수 있으며, 열차번호 인식장치는 해당 정보를 열차번호관리 프로세스에게 전송하여 열차의 위치추적에 대한 보정이 가능하도록 한다.

즉 열차번호관리 프로세스는 LDTS와 열차번호 인식장치로부터 수신되는 정보를 이용하여 보다 정확한 열차의 위치추적을 수행하며, 타 역에서 전송한 열차번호와 열차위치 정보를 활용하여 경로제어 서브시스템에 의한 열차의 자동진로제어가 가능하도록 한다.

나. 표시정보관리 프로세스

표시정보관리 프로세스는 LDTS로부터 수신되는 현장 선로변 장치들(신호기, 선로전환기, 궤도회로 등)의 물리적 표시 정보를 논리적 표시 정보로 변환하여 데이터베이스를 갱신하며, 자율분산 역제어시스템의 모든 프로세스들이 공유하여 사용가능하게 한다. 표시정보관리 프로세스는 선로변 장치들의 상태정보를 수신하기 위해 LDTS를 주기적으로 폴링 하여 데이터베이스의 상태 값을 업데이트한다.

다. 설계 및 개발

(1) 유즈케이스

(가) 열차추적

실시간 열차의 이동을 감시하며 열차번호의 이동을 조절한다.

(나) 열번관리

궤도점유 정보에 열차번호를 할당하거나 삭제한다.

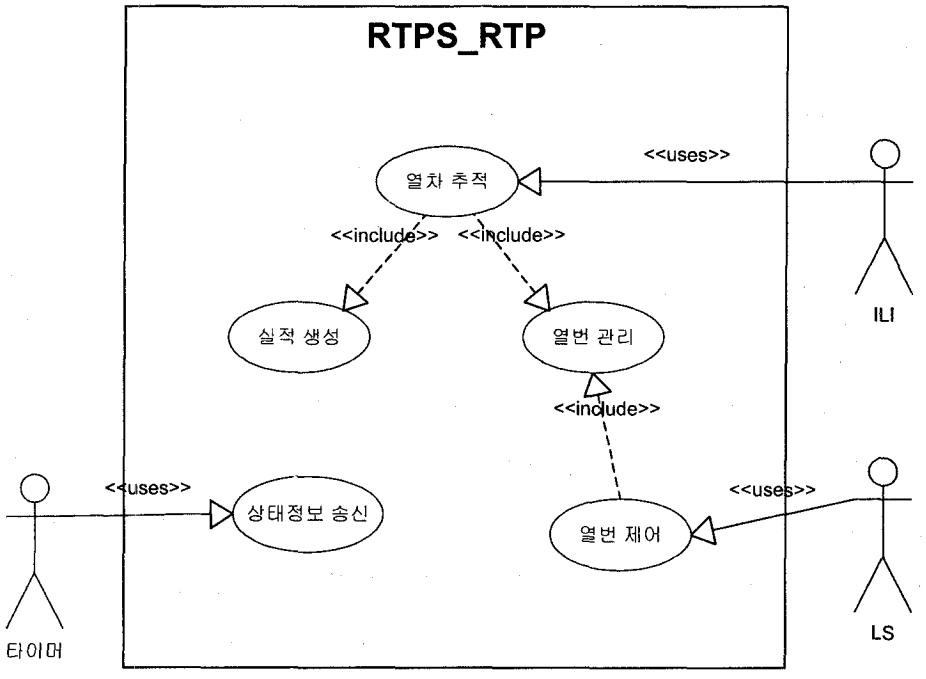


그림 3-3 RTPS 유즈케이스

(다) 실적생성

열차의 이동에 따라 출·도착 진로실적 출·도착 시간 실적을 생성 기록한다.

(라) 열번제어

L/S의 수동열차번호 제어 명령을 처리한다.

(마) 상태정보 송신

일정주기로 응용프로그램의 상태정보를 DF(Data Field)로 송신한다.

(2) 유즈케이스 명세

(가) 열차추적 유즈케이스

① 사전조건

본선 및 입환 열차스케줄이 존재해야 한다.

② 사후조건

점유정보에 열차번호가 부여되며 점유정보 이동에 따라 열차번호가 이동하게 된다.

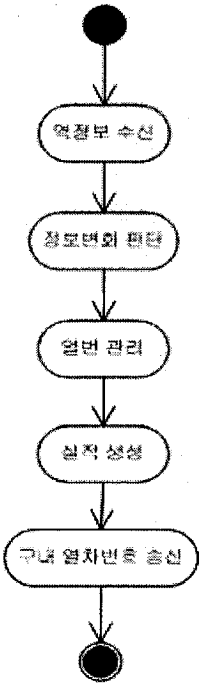


그림 3-4 열차추적

③ 기본흐름

■ 역정보 수신

ILI에서 가공된 엘리먼트 단위의 역 점유정보를 수신한다.

■ 정보변화 판단.

궤도 점유나 해정에 따라 다른 처리를 하게 된다.

■ 열차번호 생성, 이동

- 역경계 궤도에서의 궤도 점유에 의해 열차번호의 생성 이루어진다.
- 점유정보의 연속적인 이동에 의해 열차번호의 이동이 처리된다.

■ 열차번호 삭제

역경계 궤도에서의 궤도 해정에 의해 열차번호의 삭제가 이루어진다.

■ 출·도착 진로, 시간 실적 기록

진로구간 첫궤도 점유 시 출·도착 진로 실적 및 출발시간 실적이 작성된다.

홈궤도 점유 시 도착시간 실적이 작성된다.

■ 구내 열차번호 송신

생성된 열차번호는 표시 정보로 변환되어 DF에 송신된다.

④ 대체흐름

모든 점유정보를 동일하게 반복한다.

(3) 열번관리 유즈케이스

① 사전조건

본선 및 입환의 당일 스케줄이 존재한다.

② 기본흐름

■ 열번삭제

■ 열번생성

③ 대체흐름

■ 위치설정

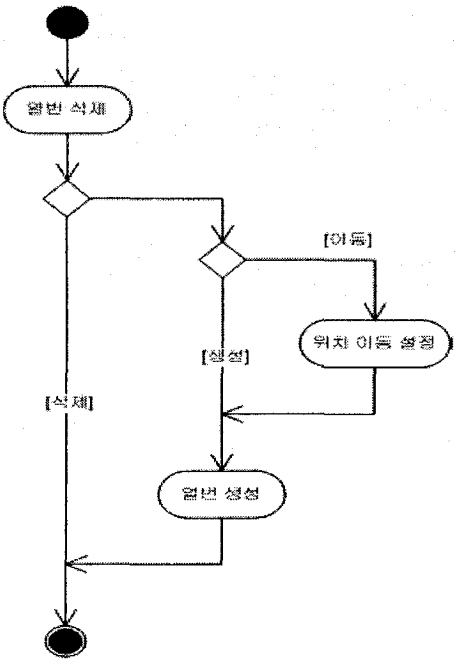


그림 3-5 열 번관리

(가) 실적생성 유즈케이스

① 사전조건

스케줄이 존재해야 한다.

② 사후조건

열차운행 스케줄 실적을 조회할 수 있다.

③ 이벤트 흐름

■ 궤도 타입 판단

홈 궤도인지 판단한다.

도착 및 출발 진로의 첫 궤도인지 판단한다.

■ 도착 진로 실적처리

장내신호 내방 첫 궤도인 경우 도착 진로 실적을 스케줄에 기록한다.

■ 도착시간 실적처리

홈궤도 점유시 도착시간 실적을 스케줄에 기록한다.

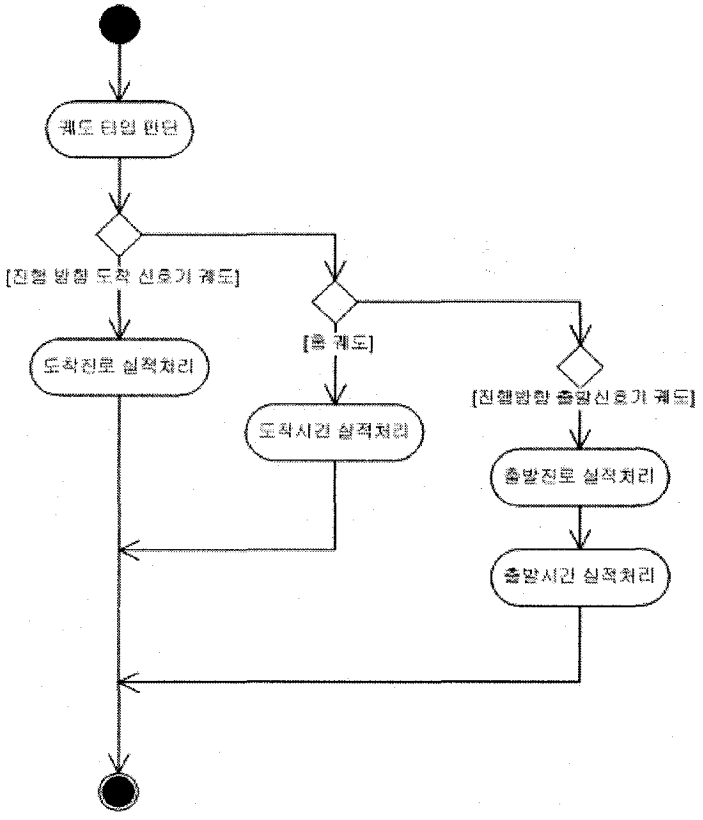


그림 3-6 실적 생성

■ 출발진로 실적처리

출발 진로 내방 첫 궤도인 경우 출발 진로 실적을 스케줄에 기록한다.

■ 출발시간 실적처리

출발 진로 내방 첫 궤도인 경우 출발시간 실적을 스케줄에 기록한다.

(나) 열번 제어 유즈케이스

① 사전조건

L/S가 수동 열번 제어 메시지를 생성한다.

② 사후조건

생성된 열차번호가 열차추적 및 표시에 사용된다.

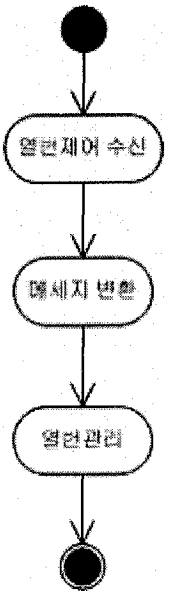


그림 3-7 열번제어

③ 기본흐름

■ 열 번 제어 수신

L/S로부터 수동 열차번호 제어 메시지를 수신한다.

■ 메시지 변환

제어메시지를 열 번 관리가 사용할 수 있도록 변환한다.

■ 열번관리

제어메시지를 열 번 관리를 이용하여 적용한다.

(4) 클래스 다이어그램

(가) TP Class

- 초기화 설정을 한다.
- 열차 점유 정보를 이용해서 열차 번호를 생성한다.
- 출고 궤도가 없는 출고 열차 번호를 자동으로 생성한다.
- 열차번호를 데이터 필드에 전송한다.
- 열차번호 수동 삽입 제어를 처리한다.
- 열차번호 수동 삭제 제어를 처리한다.

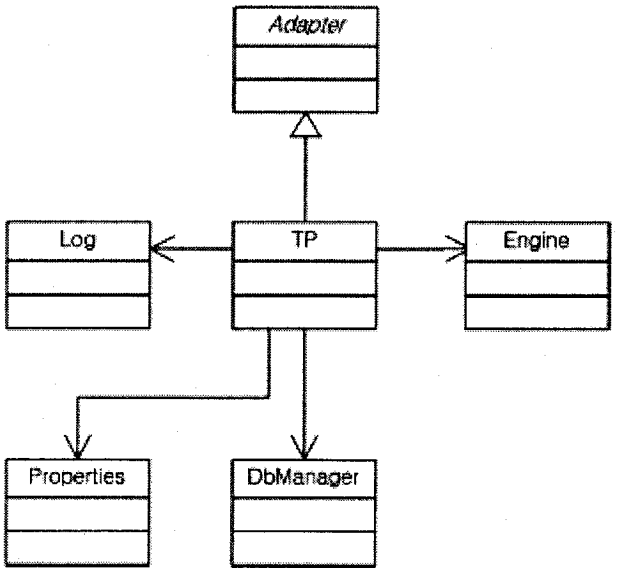


그림 3-8 TP 클래스 다이어그램

(나) Engine Class

- 열차번호를 생성한다.
- 열차번호를 삭제한다.
- 변화된 점유 정보를 판독한다.
- 스케줄을 이용해서 통과 열차를 판별한다.

- 표시에 필요한 열차번호 콘텐츠를 생성한다.
- 열차의 변화된 정보를 갱신한다.
- 열차의 변화된 위치 정보를 갱신한다.
- 열차의 시스템 진입시 열차번호를 생성한다.
- 열차의 시스템 진출시 열차번호를 삭제하다.
- 회차시 열차번호를 교환한다.
- 입환시 열차번호의 순서를 변경한다.
- 열차의 진로 내방 진입을 판단한다.
- 선로전환기가 있는 궤도의 열차번호 이동.
- 선로전환기가 없는 궤도의 열차번호 이동
- 열차의 이동 실적 기록

(5) 클래스 명세

(가) TP Class

- notifyMsg() : ADM으로부터 구동 메시지를 받는다
- notifyMsgFromLdtsComm : LDTS로부터 구동 메시지를 받는다.
- run() :쓰레드
- runOnlineAdjStMsg() : 인접역 온라인 콘텐츠 처리
- runOnlineStMsg() :당역 온라인 콘텐츠 처리
- runTestlineAdjStMsg() :인접역 테스트라인 콘텐츠 처리
- runTestLineStMsg() :당역 테스트라인 콘텐츠 처리
- waitMsg() :이벤트 대기
- loadProperties() : 응용프로그램 기본값 설정
- runRecvMessage() :수신된 콘텐츠 처리

(나) Engine Class

- createTrain() :열차번호 생성
- deleteAllTrain() :모든 열차번호 삭제
- detectChangeStatus() : 점유정보 관리
- getNewMsg() :새로운 메시지 처리
- getOldMsg() :이전메세지 처리
- isPassTrain() :통과 열차 판단
- makeTrainDisplay() :열차번호 표시정보 생성
- popShuntTrain() :출고시 무점유 열차번호 생성 처리
- updateTrain() :열차의 정보변화 갱신
- changeTrainNumber() : 열차번호 교환
- changeTrainPosition() : 열차의 위치이동 처리
- enterSystem() :열차번호 시스템 진입 처리
- exitSystem() :열차번호 시스템 진출 처리
- isRouteStartTrack() :진로내방 진입 처리
- moveNsTrack() :선로전환기가 없는 궤도 이동
- moveSwTrack() :선로전환기가 있는 궤도 이동
- moveTrain() :열차 이동
- saveTrainHistory() : 열차 이동 실적 기록
- setScheduleFlag() :진로를 구성한 스케줄 기록

라. ILI (InterLocking Interface)

(1) 유즈케이스

(가) 현장표시

연동장치 또는 연동장치 인터페이스(LDTS) 장비로부터 BIT 표시 정보를 수신하여 엘리먼트 단위의 정보로 변환 DF에 송신한다.

(나) 현장제어

L/S의 수동 제어나 RC의 자동진로 제어 메세지를 BIT제어 정보로 변환 연동장

치나 연동장치 인터페이스(LDTS) 장치에 송신한다.

(다) 상태정보 송신

ILI와 연동장치 인터페이스(LDTS) 장비의 상태정보를 DF에 송신한다.

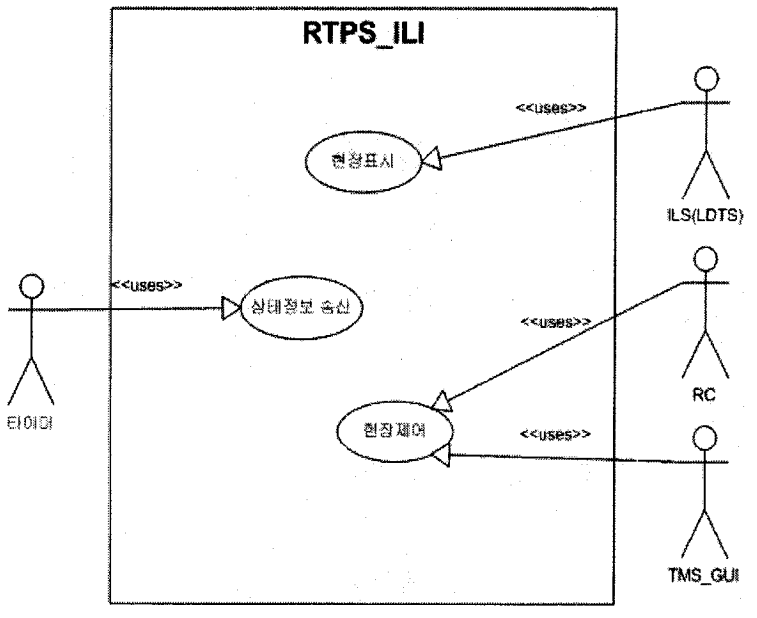


그림 3-9 ILI 유즈케이스

(2) 유즈케이스 명세

(가) 현장표시 유즈케이스

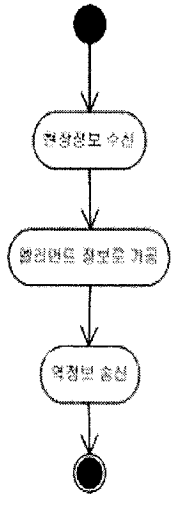


그림 3-10 현장표시

① 사전조건

연동장치 또는 연동장치 인터페이스 장비(LDTS)로부터 현장 정보를 받을 수 있어야 한다.

② 사후조건

비트단위의 현장정보는 엘리먼트 단위의 역정보로 전환된다.

③ 기본흐름

- 현장정보 수신

연동장치 또는 LDTS로부터 비트단위의 현장 정보를 받는다.

LDTS와의 인터페이스는 RS422 시리얼 통신을 사용한다.

CTC-LDTS간 프로토콜을 사용하며 LDTS 프로토콜은 지역별로 다를 수 있다.

- 엘리먼트 정보로 가공

IO리스트의 정보를 이용해 비트 정보를 바이트 또는 워드정보로 변환한다.

- 역정보 송신

엘리먼트 단위로 가공된 현장정보는 DF로 송신된다.

(나) 현장제어 유즈케이스

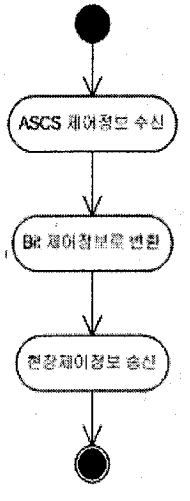


그림 3-11 현장제어

① 사전조건

연동장치 및 연동장치 인터페이스(LDTS)에 제어 정보를 전송할 수 있어야 한다.

② 사후조건

연동장치 및 연동장치 인터페이스에 현장 제어 정보를 전송한다.

③ 기본흐름

- 제어정보 수신
- BIT 제어정보로 변환
- 현장 제어정보 송신

(3) 클래스 다이어그램

(가) ILI

- 프로그램을 기동한다.
- LDTS로부터 수신된 표시 정보를 엔진클래스로 전달한다.
- 엔진클래스로부터 수신된 제어 정보를 LDTS로 전달한다.
- 온라인 데이터를 처리한다.
- 테스트라인 데이터를 처리한다.

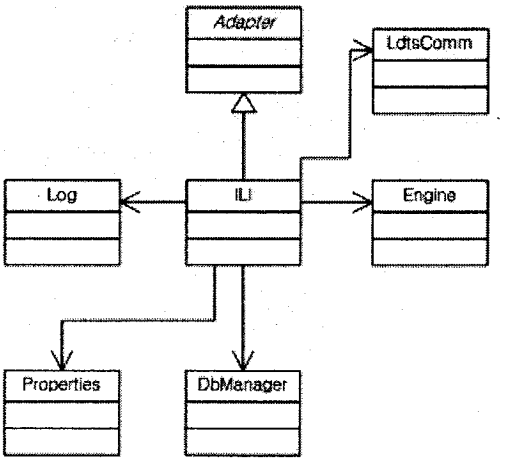


그림 3-12 ILI 클래스 다이어그램

(나) Log

- 응용프로그램의 동작상태를 텍스트로 콘솔에 출력한다.
- 콘솔 출력 레벨을 제어한다.

(다) Properties

- 통신포트 매개변수를 읽는다.
- 제어가능 여부 매개변수를 읽는다.

(라) DbManager

- 데이터 베이스를 관리한다.
- 메모리 맵 유형의 자료구조를 관리한다

(마) LdtsComm

- 통신 속도 및 포트를 설정한다.
- 쓰레드로 동작한다.
- 시리얼 데이터를 LDTS 프로토콜에 적용하여 송/수신한다.

(바) Engine

- 제어메세지를 LDTS타입으로 변환한다.
- 제어메세지를 시리얼 통신으로 전송한다.
- 현장 정보를 콘텐츠 1로 변환한다.
- 진로정보를 이용해 궤도별 진로 정보를 생성한다.
- 진로구성후 궤도 점유를 관리한다.
- 구형 LDTS프로토콜인 경우 비트정보를 가공한다.
- 진로 제어시 제어 신호기를 조사한다.
- 진로 제어시 경합 신호기를 조사한다.
- 진로 제어시 선로전환기를 조사한다.
- 진로 제어시 궤도를 조사한다.

(4) 클래스 다이어그램 명세

(가) LdtsComm Class

① 메소드

- run() : 쓰레드
- LdtsComm() :생성자
- set/get Port() :시리얼 통신 포트 속성 함수

- set/get Speed() : 시리얼 통신 속도 속성 함수
- connect() : 시리얼 통신 연결
- tryConnection() :시리얼 통신 관리

(나) ILI Class

① 메소드

- notifyMsg() : ADM으로부터 구동 메시지를 받는다
- notifyMsgFromLdtsComm : LDTS로부터 구동 메시지를 받는다.
- run() :쓰레드
- runOnlineAdjStMsg() : 인접역 온라인 콘텐츠 처리
- runOnlineStMsg() :당역 온라인 콘텐츠 처리
- runTestlineAdjStMsg() :인접역 테스트라인 콘텐츠 처리
- runTestLineStMsg() :당역 테스트라인 콘텐츠 처리
- waitMsg() :이벤트 대기
- callRouteResetTimer() :예비 진로 취소
- loadProperties() : 응용프로그램 기본값 설정
- makeSendMessage() : 콘텐츠 전송
- runRecvMessage() :수신된 콘텐츠 처리
- setRouteTrack() : 진로설정시 궤도 처리

(다) Engine Class

① 메소드

- controlAscsCommand() : 수신된 제어메세지를 처리한다.
- controlStation() :LDTS에게 제어메세지를 송신한다.
- makeBsMsg() :COS 정보를 BS정보로 변환한다.
- makeBsToCc1() : BS정보를 CC 코드1 메시지로 변환
- makeTrackOccupyHistory() :진로설정후 점유 관리
- getWspName() :WSP 식별자 생성
- getWspDir() :WSP 방향 설정

- isRouteSet() :진로 설정 유무 판단
- makeMainSignalData() :신호기 정보 생성
- makeModeData() :모드 정보 생성
- makePowerData() :전원 정보 생성
- makeShuntingSignalData() :입환 신호기 정보 생성
- makeSignalData() :신호기 정보 처리
- makeSwitchData() :선로전환기 정보처리
- makeTrackOccupyData() :궤도 점유정보 처리
- makeTrakRuteClear() : 궤도별 진로설정 취소
- searchControlSignal() : 제어 신호기 조사
- searchRoute() : 쇄정 진로 조사
- searchRouteSwitch() :진로내 쇄정 선로전환기 조사
- searchRouteTrack() :진로내 쇄정 궤도회로 조사
- searchSwitch() :쇄정 선로전환기 조사
- searchTrack() :쇄정 궤도회로 조사

마. 하드웨어 구성

표 3-1 RTPS 서브랙 구성(1개역 기준)

구분	제품 사양	수량	비고
서브랙	MODEL : ADLINK cPCIS-3320 - 19" 9U Height CPCI Platform, Dual System(8Slots + 5Slots) - 64-Bit, H.100 bus, P3/P5 rear I/O - cPS -H325/AC*4 - redundant power supply	2 SETS	DUAL POWER
CPU, M/B	MODEL: cPCI -6841/PM16/M1G/H80G - Intel Pentium M processor 1.6 GHz - Dual Ethernet on board - 1GB memory on board with ECC - Intel E7501 chipset , 2.5" HDD 80G	2 SETS	
TM	Rear I/O transition module for cPCI-6840(V)		10/100Mbps
랙용KVM	ATEN 랙용 KVM CN-1280MK - 17" LCD, 8 PORT INPUT - 1 OUTPUT	1 SET	시스템공통
O/S	Windows XP Pro	2 Copy	

2. 경로제어 서브시스템

경로제어 서브시스템(RCS : Route Control Subsystem)은 열차의 위치와 이동방향 정보를 기초로 하여 자동진로설정 논리에 의해 역에서 열차의 진로를 자동으로 제어한다.

가. 열차경로제어 프로세스

열차경로제어 프로세스는 실시간 열차위치추적 서브시스템(RTPS)의 표시정보관리 프로세스와 열차번호관리 프로세스로부터 현장 선로변 시설물의 논리적 상태정보와 열차의 이동정보(위치, 열차번호, 방향 등)를 수신한다. 수신된 정보를 이용하여 자율분산역내의 특정한 현장조건(지정된 궤도회로의 점유 등)이 만족하고 제어모드가 역 자동 제어모드인 경우, 열차스케줄로부터 해당 열차의 진입정보를 통하여 현장의 연동조건과 시설물의 상태를 판단하여 진로제어 명령을 자동으로 생성하게 된다.

자율분산역의 본선에 진입한 열차 중에서 열차조성 등의 입환 작업이 필요한 경우, 열차를 입환 스케줄과 비교하여 입환선 또는 대피선으로 자동진로 설정을 제어하게 된다. 설정되는 모든 진로제어 정보는 LDTS로 전송되며, LDTS로부터 해당 정보를 수신한 연동장치에 의해 진로가 설정된다.

나. 설계 및 개발

(1) 유즈케이스

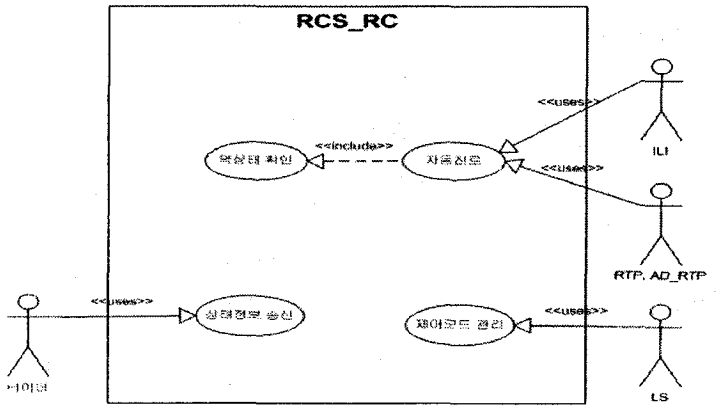


그림 3-13 RC 유즈케이스 다이어그램

(가) 역상태 확인

역정보를 활용한다.

(나) 자동진로 제어

당역 또는 인접역 열차번호 정보를 이용해 자동진로제어 모드인 경우 자동진로를 제어한다.

(다) 모드제어

L/S로부터 수신된 제어모드 정보를 이용해 자동진로제어 사용유무를 설정한다.

(라) 상태정보 송신

자동진로 응용프로그램의 상태 정보를 DF에 송신한다.

(2) 유즈케이스 명세

(가) 제어모드 관리 유즈케이스

① 사전조건

L/S에서 제어한 자동진로 제어모드 상태가 있어야 한다.

② 사후조건

자동모드인 경우 자동진로 제어가 시작된다.

③ 기본흐름

CC1에서 자동진로 모드를 확인한다.

(나) 자동진로 유즈케이스

① 사전 조건

- 열차 번호가 존재해야 한다.
- 제어 대상 열차의 스케줄이 존재해야 한다.
- 제어 모드가 자동이어야 한다.

② 사후 조건

열차의 진로제어가 자동으로 이루어져야 한다.

③ 기본흐름

- 대상 열차 리스트를 초기화 한다.
  - 대상 열차를 삭제한다.
  - HR의 제어 스케줄의 열차를 삭제한다.
  - RC의 제어 스케줄의 열차를 삭제한다.
- 대상 열차를 추가한다.
  - 열차의 스케줄이 있으면 선정한다.
  - 열차의 실적이 없으면 선정한다.
- 대상 열차의 스케줄이 HR에 의해 변경되었는지 확인한다.
  - 대상 스케줄을 HR의 제어스케줄과 비교하여 HR의 제어스케줄을 우선한다.
- 진로 제어 리스트를 작성한다.
  - 대상열차의 스케줄로 진로제어 리스트를 작성한다.
  - HR에 의해서 제어 스케줄이 변경되면 그것을 우선한다.
  - HR의 제어 스케줄에 없더라도 열차번호가 있다면 스케줄에 포함된다.

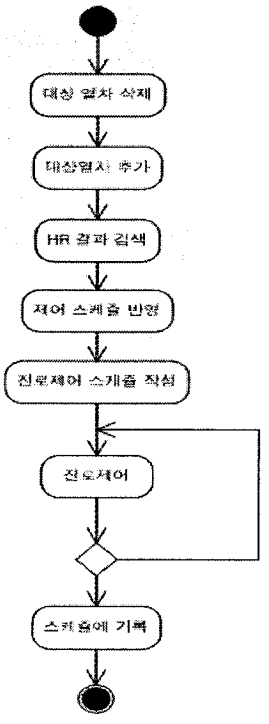


그림 3-14 자동진로

유즈케이스

- 진로를 제어한다.

  - 스케줄에 진로제어 설정이 되어 있지 않아야 한다.

- 제어가 완료되었는지 확인한다.

  - CC1의 진로설정 정보를 이용하여 제어가 완료되었는지 확인한다.

  - 동일 열차의 동일 제어 요구시 빠른 시간의 열차가 제어한 것으로 간주한다.

- 스케줄에 진로 제어를 기록한다.

  - HR에서 사용하기 위한 정보를 제어결과를 스케줄에 설정한다.

  ④ 예외흐름

- 당역 인접역 모두 동일한 처리를 한다.

- 진로가 제어되지 않았다

  제어후 현장조건에 의해 제어할 수 없다.

- 진로를 다시 제어한다.

  제어되지 않은 경우 다시 제어한다.

(3) 클래스 다어어그램

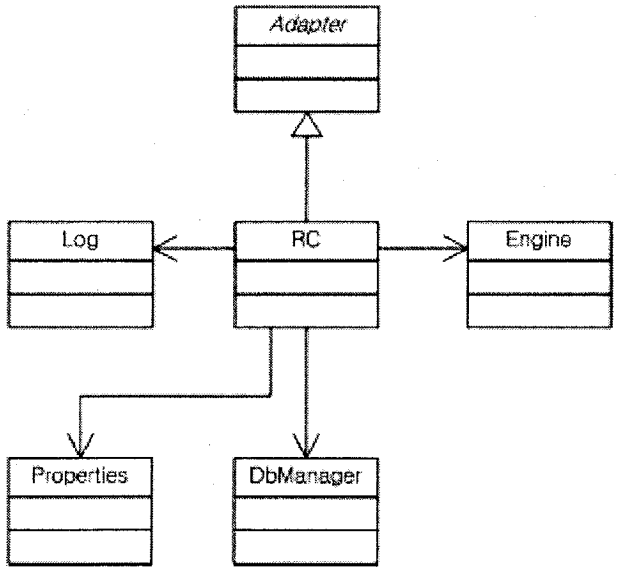


그림 3-15 RC 클래스 다이어그램

(가) RC Class

- 파라미터를 설정한다.

- 쓰레드를 사용한다.
- ADM으로부터 데이터 필드의 데이터를 수집한다.
- 진로제어 메시지를 ADM을 통해 데이터 필드에 전송한다.

(나) Engine Class

- 자동진로 대상 열차를 삭제한다.
- 자동진로 대상열차를 추가한다.
- 스케줄과 HR의 제어스케줄을 결합한다.
- 통과열차를 구별한다.
- 현장 제어를 위한 스케줄을 생성한다.
- 동일 열차 진로 제어시 선행 열차를 우선한다.
- 진로를 제어한다.
- 스케줄에 진로설정 결과를 기록한다.

(4) 클래스 명세

(가) RC Class

① 속성

- arrAlphaSec : 도착진로 제어 시간 계산 상수
- dbManager : 데이터베이스 지시자
- dptAlphaSec :출발진로 제어시간 계산 상수
- engine : engine 지시자
- shuntAlphaSec : 입환진로 제어 시간 계산 상수
- symbolStatus : 현재 심볼(엘리먼트) 상태
- appVersion : 응용프로그램 버젼
- logger :Log4j 지시자

(나) Engine Class

① 속성

- logger :Log4j 지시자

② 메소드

- RouteControl() :자동진로 제어 처리
- addTargetScheduel() :대상 스케줄 생성
- makeRcSchedule() :자동진로 스케줄 생성
- clearTargetSchedule() :대상 진로 초기화
- makeRcScheduleReorder() : 경합진로 처리

다. 하드웨어 구성

표 3-2 RCS 서브랙 구성(1개역 기준)

구분	제품 사양	수량	비고
서브랙	-. 19 inch VME64x Dual Rack -. VME64x 5 slots backplane with P0 x 2(좌/우) -. Module power supply 6u 600W x 2(좌/우) -. Power backplane, switch x 2(좌/우) -. FAN Tray x 2(좌/우) -. Rear 80mm TM slot x 2 (좌/우) -. Vent cover, handle	1 SET	DUAL M/B
CPU, M/B	MODEL : VR90F4B10DI1(RoHS Version) -. 6U VME-64 PC with Pentium M 1,8GHz, 1 slot only -. 1GB DDR SDRAM, front VGA/LCD 64MB, 2x front Gigabit-Ethernet -. 1x PMC(64bit/66MHz), Ultra ATA/100, 2x Seriel, 1x LPT, 3x USB -. Keyb/mouse/floppy I/F, RTC+battery, timer, watchdog -. 80GB HDD 2.5 inch	2 SET	VME
VX9-TM	-. 6U/80mm Depth Transition Module for VR9 -. PMC I/O, DVI-D x 1, VGAx 1, IDE x 1 -. USB x 3, LPT x 1 -. Keyboard & Mouse -. Include 2GB Compact Flash	2 SET	
O/S	Windows XP	2 Copy	

3. 열차모니터링 서브시스템

열차모니터링 서브시스템(TMS; Train Monitoring Subsystem)은 자율분산 역제어시스템에서 운영자를 위한 MMI를 지원하며, 실시간 열차위치추적 서브시스템으로부터 수신되는 선로변 시설물의 표시상태, 열차번호 이동 상태 등을 이용하여 화면상에 표출하기 위한 기능을 가진다. 선로변 시설물에 대한 장애 상태를 알리기 위한 경보 기능, 이상 발생 시의 경고 기능 등이 운영자에게 제공되며, 필요시 운영자는 자율분산 역에 대한 수동 제어를 실행할 수 있다.

열차모니터링 서브시스템은 기능의 특성에 따라 화면표시 프로세스와 현장제어 프로세스, 홈 리스케줄링 프로세스 (Home Rescheduling Process, 이하 HR) 등으로 구성된다.

가. 화면표시 프로세스

화면표시 프로세스는 실시간 열차위치추적 서브시스템의 표시정보관리 프로세스로부터 각종 선로변 시설물의 논리적인 표시정보, 열차번호관리 프로세스로부터 열차번호 정보 및 경로제어 서브시스템의 홈 리스케줄링 프로세스로부터 열차스케줄 정보(계획 또는 변경 스케줄) 등을 수신하여 화면상에 표출 또는 경보를 발생한다.

열차번호의 표시 형식은 열차번호, 열차등급, 진행방향, 운행상태(조착, 정상 또는 지연 등) 등의 정보를 포함하며, 열차 운행상태가 지연일 경우에는 모니터 상에 경고 메시지를 발생하여 운영자로 하여금 관련된 조치를 취할 수 있도록 한다.

운영자 ID및 암호는 시스템 관리자에 의하여 관리되며, 사용자에 대한 이력은 데이터베이스에 기록 저장되어 관리한다.

나. 현장제어 프로세스

현장제어 프로세스는 필요시 운영자로 하여금 현장 시설물에 대한 수동제어를 실행할 수 있도록 하며, 단동제어(선로전환기 제어) 또는 연동제어(진로 제어)를 할 수 있도록 메뉴 창을 제공한다. 운영자에 의한 수동제어 명령은 로그 파일에 기록 저장되며, 필요시 열람하여 기록된 내용을 확인할 수 있지만 기록된 내용은 수정 또는 변형할 수 없도록 별도의 툴을 이용하여 압축, 암호화된다.

다. 홈 리스케줄링 프로세스

(1) 홈 리스케줄링 프로세스 업그레이드

홈 리스케줄링 프로세스(이하 HR)는 해당역의 열차운행 관리 최적화를 위한 프로세스로 제어가능한 구간이 물리적 범위가 되며 인접역에서 들어오는 운행실적을 바탕으로 일정시간 내에 해당 역에서 발생하는 경합을 검지하고 출발시간, 출발/도착 진로를 변경하여 열차지연을 최소화하는 변경스케줄을 경로제어 서브시스템으로 전송한다.

(가) 홈 리스케줄링 열차스케줄 정의

① 홈 리스케줄링 열차스케줄 전이도

- 결합스케줄(본선스케줄, 입환 스케줄)을 가져온다.
- 결합스케줄을 이용해서 HR 대상열차 리스트를 생성한다.
- HR 대상열차 리스트에 있는 열차에 대해서 기본스케줄과 제어스케줄을 생성한다.
- RTPS로부터의 열차위치정보(인접역 RTPS), 연동정보와 결합스케줄, 출/도착 실적정보(해당역 RTPS) 등을 이용해서 지정한 열차에 대한 예측스케줄을 작성한다.
- 예측스케줄로부터 경합해소 스케줄을 생성한다.
- 경합해소 스케줄을 이용하여 제어스케줄을 수정한다.

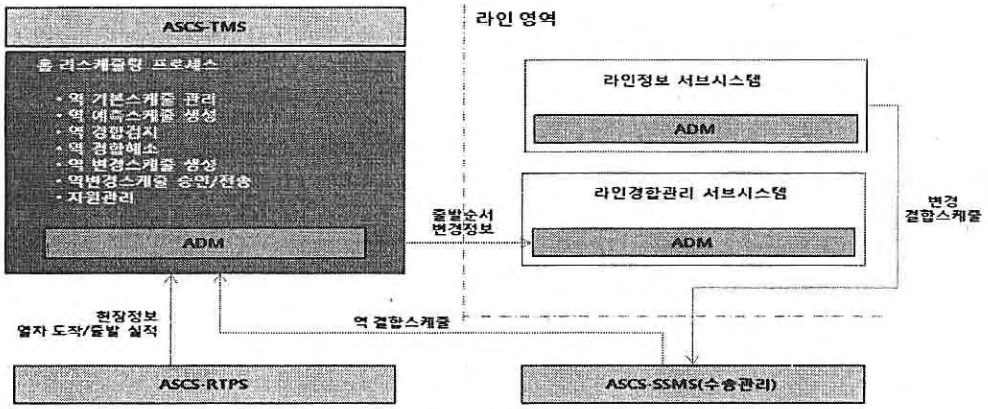


그림 3-16 홈 리스케줄링 프로세스 기본설계

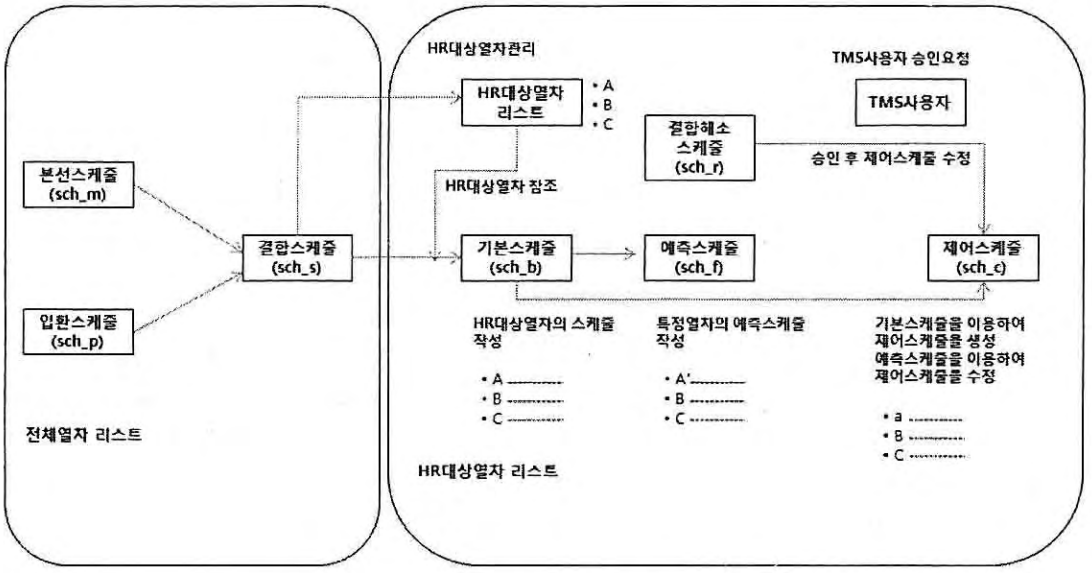


그림 3-17 HR 열차스케줄 전이도

② 결합스케줄

○ 결합스케줄은 본선스케줄과 입환 스케줄을 결합한 스케줄로, HR은 파라메타로 설정한 주기마다 결합스케줄을 가져온다.

③ HR 기본스케줄

○ 기본 스케줄은 결합스케줄 중에서 HR 대상열차에 포함된 열차스케줄만을 별도로 저장한 스케줄이며, 데이터베이스의 속성은 결합스케줄에서 실적부분을 뺀 부분과 같다.

④ HR 예측스케줄

○ 예측 스케줄은 기본스케줄 중에서 변경된 열차위치정보와 연동정보를 바탕으로 "열차도착 및 출발시간 예측로직"을 이용해서 예측한 스케줄로, HR 대상 열차에 포함된 열차스케줄만으로 구성되며, 데이터베이스의 속성은 기본스케줄과 같다.

⑤ HR 제어스케줄

○ 제어 스케줄은 기본 스케줄을 기반으로 생성된 스케줄이며, 데이터베이스의 속성은 기본스케줄과 같다. 경합해소 결과로, 경합해소 스케줄이 생성되면 제어스케줄을 변경한다.

⑥ HR 대상열차 리스트

○ HR대상열차 리스트는 결합스케줄 중에서 경합의 대상이 되는 열차를 기록한 데이터베이스이다. HR이 경합을 검지하고 경합을 해소할 때, 시간 조정을 변경하지 않을 필요가 있기 때문에 진로를 변경하지 못한다.

○ HR 대상열차 리스트에서 해당 열차의 출발/도착 시간 및 진로를 변경하지 않을 경우는 다음과 같다.

- 열차가 진로/시간 수정불가 범위 일 때
- 이미 진로설정이 완료되었을 때(통과열차의 경우)

(나) 기능 정의

① 메시지 송수신

- 외부장치로의 데이터 송신

○ 결합스케줄을 경로제어서브시스템으로 송신한다.

- 외부장치로부터의 데이터 수신

○ ADM을 통해서 인접역 RTPS로부터 열차의 위치정보(출발/통과)를 CC형태로 수신한다.

○ HR에 영향을 끼치는 연동정보를 CC형태로 수신한다.(열차고장, 궤도고장, 신호기 고장, 선로전환기 고장 등)

○ 결합스케줄에 진로설정 정보가 있으면, HR이 경합 해소 시에, HR 대상열차의 진로를 변경하지 못하도록 설정한다.

o TMS내의 ADM자체에서 발생한 시스템 종료 정보를 수신한다.

② 경보/이벤트 관리

o HR 운영 중에 발생하는 각종 경보 및 이벤트를 기록하고 관리하는 기능이다. 세부 처리 내용은 다음과 같다.

- 경보/이벤트 파라메타 파일로부터 경보/이벤트 설정 정보를 읽어온다.
- 내부 메일박스로부터 데이터를 수신한다.
- 수신한 데이터를 이용하여 경보, 이벤트의 문구를 생성한다.
- 수신한 데이터와 문구를 데이터베이스에 기록한다.
- 생성한 경보/이벤트 문구를 화면에 표시하도록 관련(사용자 인터페이스) 서브프로세스로 전송한다.

③ 서브프로세스 관리

o 프로세스 파라메타 파일로부터 사용할 프로세스 정보를 구한다.

o 타이머를 구동하여 주기적으로 해당 서브프로세스의 작동상태를 검사한다.

o 해당 프로세스가 비구동 중이면, 해당 프로세스를 구동시킨다.

o 시스템 종료 메시지를 수신하면, 모든 서브프로세스를 중지시킨다.

o 각각의 오류 상황에 대해서 해당 경보를 발생하여, 경보관리 서브프로세스로 해당 경보내용을 전송한다.

④ HR 대상열차 추가/삭제/수정

o 다음의 조건이 만족될 때 결합스케줄 데이터베이스로부터 해당열차 정보를 HR대상열차 데이터베이스에 추가한다.

- 인접역 RTPS로부터 열차 출발/통과 정보를 CC형태로 수신하고, HR대상열차 리스트에 해당 열차가 없을 때
- 결합스케줄에서 일정시간 내 해당역에 출도착하는 열차스케줄을 발견했을 때

o 다음의 조건이 만족될 HR대상열차 데이터베이스에서 해당열차를 삭제한다.

- 본선열차 중에서 출발/도착 실적 정보가 있으면 일정시간 후에 HR대상열차 리스트에서 해당열차를 삭제한다.
- 입환 열차 중에서 출발실적이 있으면 일정시간 후에 HR대상열차 리스트에서 해당열차를 삭제한다.

- 대상열차 삭제시 경보를 발생하여, 경보관리 서브프로세스로 해당 경보내용을 전송한다.
- 대상열차 삭제시, 기본/예측/제어스케줄 데이터베이스에서 해당 열차를 삭제한다.

○ 다음의 조건이 만족될 때, HR대상열차 데이터베이스에 있는 해당열차 정보를 수정한다.

- 결합스케줄에 진로설정 정보가 있으면, HR 대상열차 리스트에 있는 진로 플래그를 불가능으로 변경한다.
- 결합스케줄에 진로해정 정보가 있으면, HR 대상열차 리스트에 있는 진로 플래그를 가능으로 변경한다.
- HR 대상열차의 정보가 수정되면, 해당 경보를 발생하여 경보관리 서브프로세스로 해당 경보내용을 전송한다.

⑤ 기본스케줄 레코드 추가/ 삭제/ 수정

○ 결합스케줄로부터 해당열차가 HR대상열차 리스트에 추가될 때 해당 열차정보를 기본스케줄 데이터베이스에 추가된다.

○ HR대상열차 추가 정보가 있으면, 결합스케줄로부터 해당열차의 스케줄을 기본스케줄에 추가한다.

○ 기본스케줄 레코드가 변경되면, 경보를 발생하여 경보관리 서브프로세스로 해당 경보내용을 전송한다.

○ HR 대상열차가 삭제되면, 기본스케줄 데이터베이스에서 해당 열차의 스케줄을 삭제한다.

○ 경보를 생성하여 경보관리 서브프로세스로 해당 경보내용을 전송한다.

⑥ 예측스케줄 레코드 추가/ 삭제/ 수정

○ HR 대상열차 데이터베이스에 대상열차 추가 정보가 있으면, 기본스케줄로부터 예측스케줄을 생성한다.

○ 계산된 출발/도착 예측시간이 기본스케줄의 출발/도착 예정시간과 다르면, 예측스케줄 데이터베이스에서 출발/도착 시간필드를 계산된 출발/도착 예측시간으로 수정한다.

- 수정한 예측스케줄 데이터를 경합검지에 사용한다.
- 예측스케줄 데이터베이스 레코드가 변경되면, 경보를 발생하여 경보관리 서브프로세스로 해당 경보내용을 전송한다.
- HR대상열차 데이터베이스에 대상열차 삭제 정보가 있으면, 예측스케줄로부터 해당열차스케줄을 삭제한다.
- 예측스케줄 데이터베이스 레코드가 삭제되면, 경보를 발생하여 경보관리 서브프로세스로 해당 경보내용을 전송한다.
- 인접역 RTPS로부터 열차위치정보(출발/통과)를 CC형태로 수신하거나, HR대상열차 및 기본스케줄 데이터베이스 수정 후에 예측스케줄 데이터베이스를 수정한다.
- 인접역 정보를 CC에서 수신 시 "열차도착 및 출발시간 예측로직"을 이용하여 출발/도착 예측시간을 계산한다.
- 계산된 출발/도착 예측시간이 기본스케줄의 출발/도착 예정시간과 다르면, 예측스케줄 데이터베이스에서 출발/도착 시간필드를 계산된 출발/도착 예측시간으로 수정한다.
- 수정된 예측스케줄 데이터베이스는 경합검지에 사용된다.
- MMI로부터 사용자의 수정요구가 있으면, 해당 레코드를 수정한다.(열차고장, 선로고장 등의 장애 발생 시 수동 조작)
- 예측스케줄 데이터베이스 레코드가 변경되면, 경보를 발생하여 경보관리 서브프로세스로 해당 경보내용을 전송한다.

⑦ 제어스케줄 레코드 추가/ 삭제/ 수정

- HR 대상열차 데이터베이스에 대상열차 추가 정보가 있으면, 기본스케줄로부터 제어스케줄을 생성한다.
- HR대상열차가 삭제되었다는 정보가 있을 때 제어스케줄 데이터베이스에서 해당 스케줄을 삭제한다.
- 경보를 발생하고 경보관리 서브프로세스로 해당 경보내용을 전송한다.
- 경합해소를 실행한 후에 경합해소 스케줄이 변경된 경우, 제어스케줄의 내용을 경합해소 스케줄로 수정하며, 세부기능은 다음과 같다.

o 기본 스케줄이 변경되면, 제어스케줄 정보를 수정한다.

o 경합해소 스케줄이 생성되고, TMS 사용자가 승인하면 경합해소 스케줄 정보를 이용하여 제어스케줄의 수정한다.

o 경보를 발생하고 경보관리 서브프로세스로 해당 경보내용을 전송한다.

⑧ 경합검지

o HR대상열차에 한해서 경합상황이 있는지 검사한다. 경합을 검지하면 경합정보를 생성하여 MMI에 표시할 수 있도록 한다.

o 경합검지 정보는 다음과 같다.

- HR 대상열차 정보(열차번호, 출/도착 시간 변경 FLAG, 출/도착 진로 변경 FLAG)
- 기본스케줄의 데이터베이스 정보
- 예측스케줄의 데이터베이스 정보
- 경합검지 결과(순서준수 가능여부, 한계시간 초과 여부, 계산한 지연시간 등)

o 경합검지의 세부기능은 다음과 같다.

- 기본스케줄 레코드가 변경되었다는 정보가 있으면 경합검지를 수행한다.
- 예측스케줄이 변경되었다는 정보가 있으면 경합검지를 수행한다.
- 플랫폼 정보가 변경되었을 때 경합검지를 수행한다.
- 연동진로정보가 변경되었을 때 경합검지를 수행한다.
- 경합상황을 검지하면, MMI에 경합검지 정보를 표시하도록 하고, 경합해소를 수행하도록 메시지를 생성한다.
- 기본스케줄과 예측스케줄을 이용하여 열차지연시간을 계산한다.
- 열차지연시간이 한계수준을 초과하는지 검사한다.
- 계산한 열차지연시간이 한계 값을 초과하면, MMI에 해당 경합정보를 전송하고 경합해소를 실행한다.
- 경합검지 정보와 경합검지 결과를 MMI에 표시하도록 메시지를 생성한다.
- 경합이 검지되면, 경합해소를 수행하도록 한다.
- 경합검지에 대한 경보를 발생하고 경보관리 서브프로세스로 해당 경보내용을 전송한다.
- 열차의 출/도착시간 실적정보가 일정시간까지 생성되지 않으면 경보를 발생한

다.

⑨ 경합해소

o 경합해소 시 MMI에 표시할 정보는 경합해소의 결과로써 산출되는 경합해소 스케줄 정보이며, 그 내용은 다음과 같다.

- 제어스케줄 데이터베이스 정보
- 제어스케줄의 승인 여부

o 기본스케줄과 예측스케줄을 이용하여 계산한 열차지연시간이 한계 값을 초과했을 경우에, 경합해소 라이브러리를 이용하여 열차지연시간을 감소시키는 제어스케줄을 생성한다.

o 경합해소를 위해 사용자가 입력 할 수 있는 정보는 다음과 같다.

- 대안진로를 추가할 열차번호
- 본선열차의 대안 진입/진출 진로
- 입환 열차의 대안 진입/진출 진로(신호기)
- 본선 및 입환 열차에 대한 스케줄 정보

o 생성한 스케줄을 MMI화면에 표시하도록 한다.

o 제어스케줄 생성에 대한 경보를 생성하고, 경보관리 서브프로세스로 해당 경보내용을 전송한다.

o 제어스케줄에서 열차의 순서가 변경되었는지 검사한다.

o 열차순서가 변경되지 않았으면, TMS사용자에게 제어스케줄의 승인을 요구한다.

o 열차순서가 변경되었으면, CC형태로 순서변경 정보를 라인정보 서브시스템에 전송한다. 열차순서변경 정보를 MMI에 표시하고, 제어스케줄을 TMS사용자에게 승인 요청한다.

o TMS사용자의 승인 후에, 제어스케줄을 결합스케줄에 반영한다.

o 제어스케줄의 승인 여부에 대한 경보를 생성하고, 경보관리 서브프로세스로 해당 경보내용을 전송한다.

⑩ 제어 스케줄 전송

o 제어스케줄에 있는 상/하행 라인별 진입순서와 출발순서가 기본스케줄과 비

교하여 변경이 있는지를 검사한다. 처리내용은 다음과 같다.

- 제어스케줄과 기본스케줄간에 출발 순서변경이 있으면, MMI화면을 통해서 TMS사용자가 제어스케줄을 승인 할 수 있도록 MMI화면에 정보를 표시하도록 한다. 이때 제어스케줄의 순서변경 정보를 함께 표시하도록 한다.
- 제어스케줄에 진출입 순서가 변경이 있는지 검사한다.

⑪ 사용자 인터페이스 (MMI)

○ 경합검지 및 해소에 대한 정보를 표시한다.

○ HR을 운영 중에 발생한 각종 경보 및 이벤트를 표시한다.

○ 발생한 경보 및 이벤트의 확인을 수행하며, 데이터베이스에 저장된 내용을 조회할 수 있는 수단을 제공한다.

○ HR대상열차 리스트 및 스케줄 정보를 관리하는 화면을 제공한다.

○ 대상열차의 출/도착 시간과 진로정보를 관리하는 화면을 제공한다.

○ 기본스케줄을 조회하는 화면을 제공한다.

○ 예측스케줄을 조회하고, 예측스케줄을 수동으로 변경하는 화면을 제공한다.

○ 경합검지 정보를 수신하면, 해당 열차의 기본스케줄과 예측스케줄을 화면에 표시한다.

○ 경합검지 및 경합해소 관리 화면을 제공한다.

○ 진로제어 모드(자동 또는 수동)를 관리하는 화면을 제공한다.

○ 대안진로를 관리하는 화면을 제공한다.

○ MMI가 수신하여 처리하는 데이터의 종류는 다음과 같다.

- HR운영 중에 각 서브프로세스가 전송한 경보 메시지.
- 경합 검지 시 경합검지 내용.
- 진출입 순서 준수 및 지연시간 계산 결과.
- 경합해소 실행 결과 및 제어스케줄 표시요구 정보

○ MMI가 HR 데이터베이스 기록 내용은 다음과 같다.

- 연동 데이터베이스에 고장 내용 기록
- HR 대상열차 데이터베이스에 열차 추가(20분 이상 장기정차 시)

- 예측스케줄 데이터베이스에 변경정보 기록(시간 및 대안진로)
- 시스템 데이터베이스에 환경설정 파라메타 기록(최대 대상열차 수, 지연한도, 플랫폼 개수, 지연시간 최소화 방법, 열차종별 가중치, RCS 적용시간 등)
- 경보 데이터베이스에 경보 추가
- 이벤트 데이터베이스에 경보 추가
- 경보/이벤트의 확인 여부

(다) 홈 리스케줄링 인터페이스

① 사용자와의 인터페이스 (MMI)

- 경합 검지 및 해소에 필요한 각종 정보를 조회하고 관리하도록 화면을 제공한다.
- TMS자체에 있는 GUI로부터 TMS모드를 수신한다.

② 수송관리 서브시스템(SMS)과의 인터페이스

- 결합스케줄을 가져온다.
- 결합스케줄로부터 열차실적정보를 읽고, 실적정보의 상태에 따라 HR 기본/예측/제어스케줄을 관리한다.
- 해당역에서 출발하는 결합스케줄 정보를 읽는다.

③ 실시간 열차위치추적 서브시스템(RTPS)과의 인터페이스

- CC의 형태로 인접역의 열차위치정보(출발/통과)를 수신한다.
- CC의 형태로 해당역의 현장정보(고장정보)를 수신한다.

④ 진로제어 서브시스템(RCS)과의 인터페이스

- TMS사용자의 수동제어시 CC 형태로 TMS모드 정보(수동)를 전송한다.
- 변경스케줄 정보를 송신한다.

(라) 서브프로세스 정의

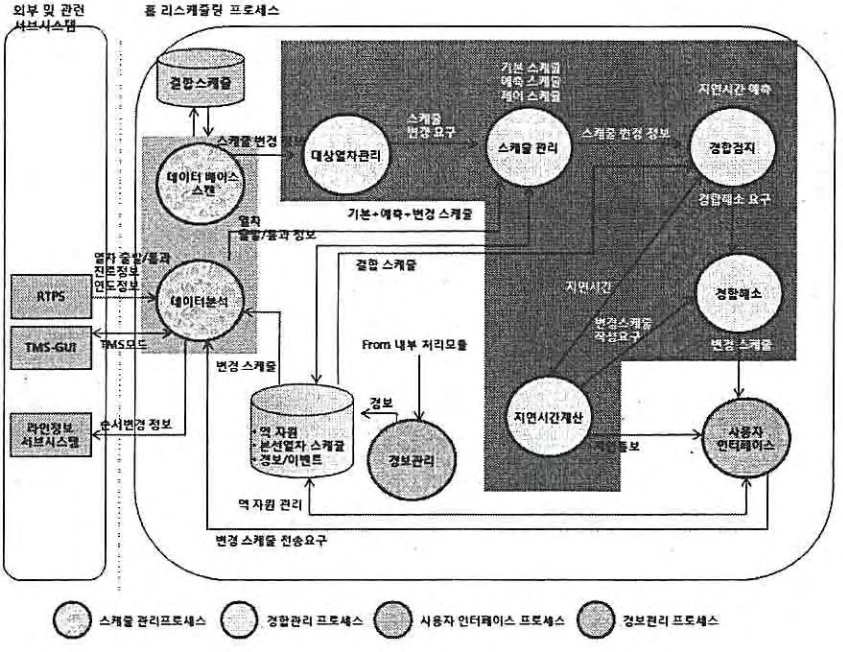


그림 3-18 홈 리스케줄링 프로세스 정의

- 4개의 프로세스를 이용하여 홈 리스케줄링의 기능을 수행한다.

표 3-3 홈 리스케줄링 프로세스 기능

ID	서브프로세스	이름	기능
1	스케줄 관리	HRProc	· 데이터베이스 스캔, 대상열차 관리, 스케줄 관리, 경합검지, 경합해소, 고장처리
2	데이터 분석	HRPars	· CC 데이터 분석 후 서브프로세스로 전송
3	사용자 인터페이스	HRMMI	· 사용자 인터페이스
4	경보관리	HRAlarm	· 경보관리

① 스케줄 관리 서브프로세스

- DFD

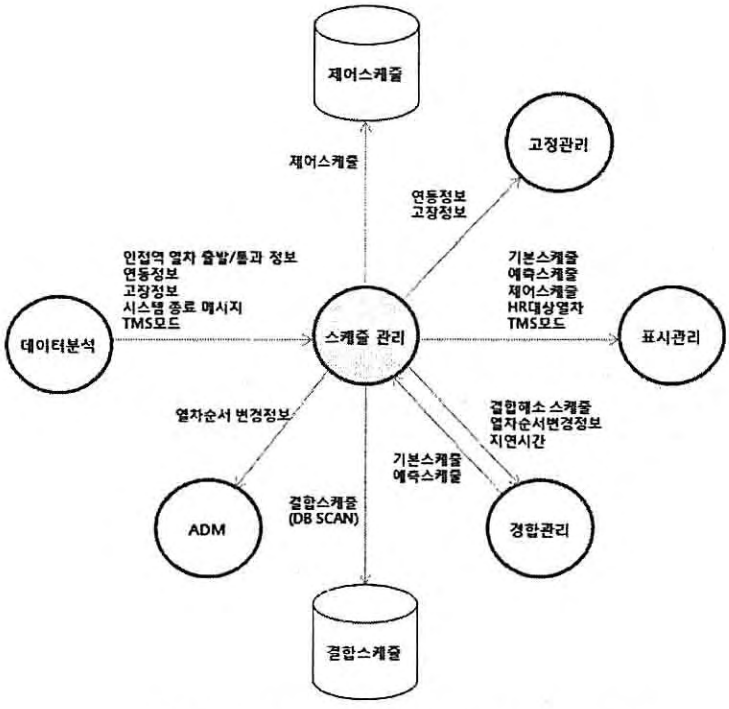


그림 3-19 스케줄 관리 서브프로세스 DF

- 서브프로세스의 세부기능은 표 3-4과 같다.

표 3-4 스케줄 관리 서브프로세스 설명

기능	설명
데이터 분석	· HR의 서브프로세스로, ADM에서 패킷 형식으로 받은 정보를 분석해서 스케줄 관리 서브프로세스로 전송하는 기능을 수행한다. · ADM에서 수신하는 정보는 인접역 열차의 출발/통과 정보, 연동정보, 시스템 종료 메시지, TMS 모드이다.
스케줄 관리	· 결합 스케줄에서 DB 스캔 방식으로 데이터를 가져온다. · 결합 스케줄을 이용해서 HR 대상열차, 기본스케줄, 예측스케줄, 제어스케줄을 생성한다. · 기본스케줄, 예측스케줄, 제어스케줄, HR 대상열차, TMS 모드, 경합정보를 표시관리로 보낸다. · 데이터 분석에서 수신한 고장정보를 고장관리로 보낸다. · 열차순서 변경 정보를 ADM으로 전송한다. · 제어스케줄을 적용한다.
경합관리	· 기본스케줄, 예측스케줄을 이용하여 경합검지를 수행한다.

	· 제어스케줄을 생성한다.
표시관리	· 스케줄 관리 서브프로세스가 처리중인 기본스케줄, 예측스케줄, 제어스케줄, HR 대상열차, TMS 모드, 경합정보를 메일박스를 통하여 MMI 서브프로세스로 전송한다.
고장관리	· 열차, 궤도, 선로전환기, 신호기의 고장/해소 정보를 수신한다. · 열차, 궤도, 선로전환기, 신호기를 포함하고 있는 진로 정보를 찾아 고장/해소 상태를 적용한다.

② 데이터 분석 서브프로세스

- DFD

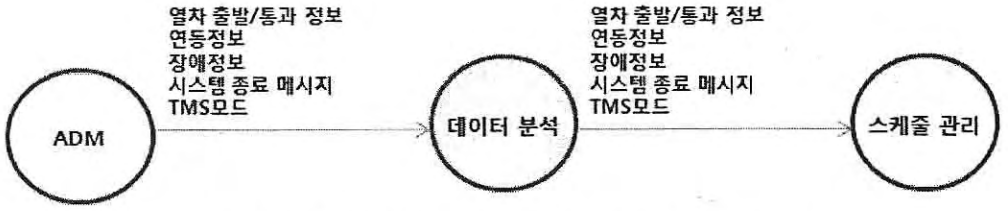


그림 3-20 데이터 분석 서브프로세스 DFD

- 서브프로세스의 세부기능은 표 3-5과 같다.

표 3-5 데이터 분석 서브프로세스 설명

기능	설명
ADM	· 열차 출발/통과 정보, 연동정보, 장애정보, TMS 모드 정보를 주기적으로 전송한다.
데이터분석	· 패킷 형식으로 ADM 데이터를 수신한다. · 패킷 데이터를 분석해서 수신한 정보가 열차 출발/통과 정보, 연동정보, 장애정보, TMS 모드 정보인지를 파악한다. · 분석된 정보가 열차 출발/통과 정보, 연동정보, 장애정보, TMS 모드 정보가 아니면 수신된 데이터를 버린다. · 분석된 정보가 열차 출발/통과 정보, 연동정보, 장애정보, TMS 모드 정보이면 데이터 항목별로 가장 최근 정보를 저장한다. · 분석된 정보와 이전에 저장된 값을 비교해서 수정된 부분이 있으면 스케줄 관리 서브프로세스로 전송한다. · ADM 에서 시스템 종료 메시지를 수신하면 스케줄 관리 서브프로세스로 전송한다.

③ 사용자 인터페이스 서브프로세스

- DFD

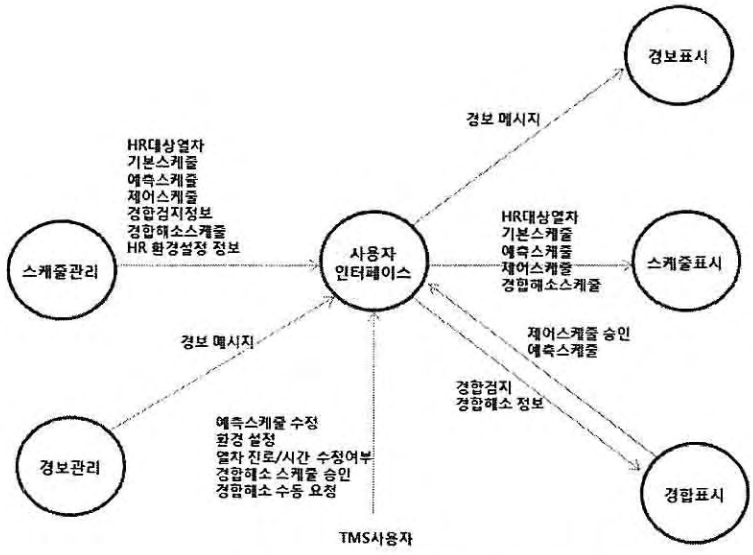


그림 3-21사용자 인터페이스 DFD

- 서브프로세스의 세부기능은 표 3-6과 같다.

표 3-6 사용자 인터페이스 설명

기능	설명
스케줄 관리	· HR 대상열차, 기본스케줄, 예측스케줄, 제어스케줄, 경합검지정보, 경합해소스케줄, HR 환경설정 정보를 사용자 인터페이스로 전송한다.
사용자 인터페이스	· 스케줄 관리에서 HR 대상열차, 기본스케줄, 예측스케줄, 제어스케줄, HR 환경설정 정보를 수신 시 스케줄 표시로 정보를 보낸다. · 스케줄 관리에서 경합검지정보, 경합해소 스케줄 정보를 수신 시 경합 표시로 정보를 보낸다. · 경보관리에서 경보 메시지를 수신 시 경보표시로 정보를 보낸다.
경보 표시	· 경보관리에서 수신된 메시지를 MMI의 알람 표시 영역에 정보를 출력한다.
스케줄 표시	· 스케줄 관리에서 기본스케줄, 예측스케줄, 제어스케줄 정보를 수신 시 MMI의 스케줄 표시 영역에 정보를 출력한다.
경합 표시	· 스케줄 관리에서 경합정보를 수신 시 MMI의 경합표시 영역에 정보를 출력한다.

④ 경보관리 서브프로세스

- DFD

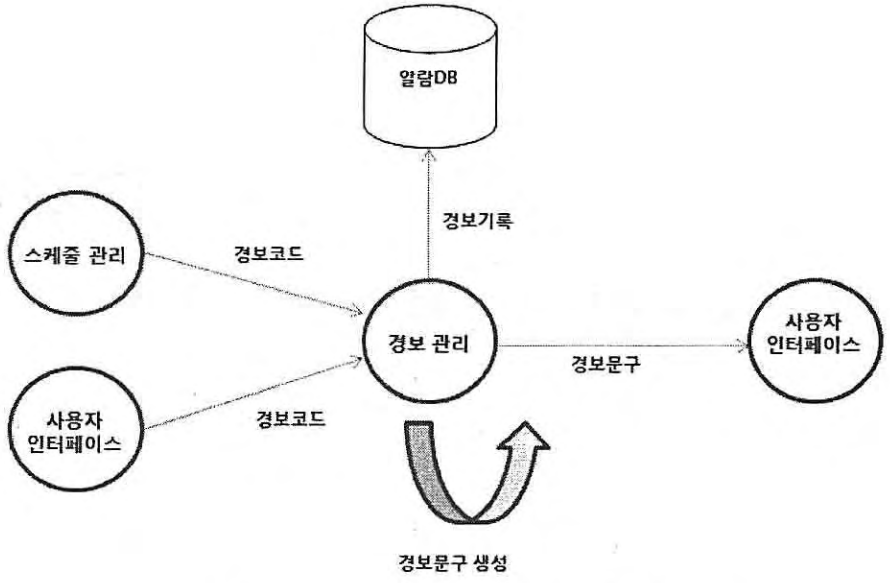


그림 3-22 경보관리 서브프로세스 DFD

- 서브프로세스의 세부기능은 표 3-7과 같다.

표 3-7 경보관리 서브프로세스 설명

기능	설명
경보 관리	· 경보 생성을 위한 정보를 DB에서 읽어온다. · HR의 서브프로세스로부터 경보 코드를 수신한다. · DB에서 읽은 정보와 경보 코드를 이용하며 경보 메시지를 생성한다. · 생성된 경보 메시지를 알람 DB에 기록한다. · 생성된 경보 메시지를 사용자 인터페이스로 전송하여 MMI에 표시되도록 한다.

(마) 경보/이벤트 정의

① 경보/이벤트 관리

사용자 인터페이스 서브프로세스는 경보관리 서브프로세스가 타 서브프로세스로부터 수신한 경보를 분석하여 만든 경보문구를 수신하여, MMI화면의 "이벤트 로

그"에 표시한다.

- 경보문구 구성

발생한 경보는 발생시각, 경보종류, 등급 그리고 내용으로 구성된다.

경보문구 : 발생시각 -> 내용

표 3-8 경보문구 구성

항목	내용	크기(Byte)
발생시각	"YYYY/MM/DD hh:mm:ss"의 24시 형식으로 표시한다.	19
내용	텍스트	120

- 경보 표시 방법

  - 경보의 배경 화면은 검정색으로 표시한다.
  - 경보의 글씨는 노란색으로 표시한다.
  - 가장 최근의 경보를 리스트 박스의 하단에 표시한다.
  - 경보가 제한된 수만큼 차면, 이전의 표시된 경보를 지우고 새로 발생하는 경보를 표시한다.

② 경보/이벤트의 종류 및 등급

- 경보의 종류 및 등급

경보는 HR 운영 중에 발생하는 각종 장애나 오류를 나타나며, 주요 경보내용은 다음과 같다.

표 3-9 이벤트의 종류

분류	종류
스케줄	· CC(Content Code) 데이터 수신 · HR 대상열차 추가, 삭제, 수정 · 기본스케줄 추가, 삭제, 수정 · 예측스케줄 추가, 삭제, 수정 · 제어스케줄 추가, 삭제, 수정

경합	· 경합검지 · 경합해소
취급 조작	· 예측스케줄 수정 · 데이터베이스 환경 설정 · HR대상열차 속성 변경 · 대안진로 설정 · 제어스케줄 승인 · 경합해소 수동 요청 · 기타 중요 MMI 조작 사항
운영	· 서브프로세스 구동

- 이벤트의 종류

이벤트는 HR운영 중에 발생하는 경보를 제외한 처리 상태나 사건 및 사용자 취급을 나타내며, 주요 이벤트의 내용은 다음과 같다.

표 3-10 경보 종류 및 등급

분류	종류	등급
스케줄	· HR 대상열차 추가, 삭제, 수정 오류 · 기본스케줄 추가, 삭제, 수정 오류 · 예측스케줄 추가, 삭제, 수정 오류 · 제어스케줄 추가, 삭제, 수정 오류 · 실적시간 초과 오류	Major
경합	· 경합검지 오류 · 경합해소 오류 · 열차출발순서변경 · 지연시간 초과	Major
취급 조작	- 취급 오류	Minor
운영	· 서브프로세스 DOWN · 통신(NETWORK) 오류	Block
	· 송수신 메시지 오류	Major
고장	· 신호설비 고장 정보(신호기, 궤도, 선로전환기, 차량 등)	Major

(2) 홈 리스케줄링 프로세스의 ASCS 통합

홈 리스케줄링 프로세스는 열차모니터링 서브시스템(TMS)의 화면표시 프로세스, 현장제어 프로세스와 마찬가지로 TMS의 프로세스로서 공용 플랫폼으로 JAVA VM을 사용하고 있다. 그러므로 홈 리스케줄링 프로세스의 구동 및 종료는 JAVA 형태로 되어있으며 HR내의 서브프로세스를 관리하고 실행하는 것은 C++형태로 구성하였다. 홈 리스케줄링은 다음과 같은 방법으로 ASCS와 통합하였다.

- HR DLL은 JNI를 이용하여 TMS내의 ADM과 인터페이스 한다.
- HR DLL 구동시 HR의 서브프로세스들을 구동시키도록 HR 관리 프로세스에 메시지를 보낸다.
- HR관리프로세스는 프로세스 구동메시지를 수신시 HR 서브프로세스를 구동시킨다.
- ADM으로부터 시스템 종료정보를 수신시 HR DLL은 HR관리 서브프로세스에 메시지를 보내 서브프로세스를 종료시킨다.
- HR 관리 서브프로세스는 서브프로세스의 구동 상태를 관리한다.

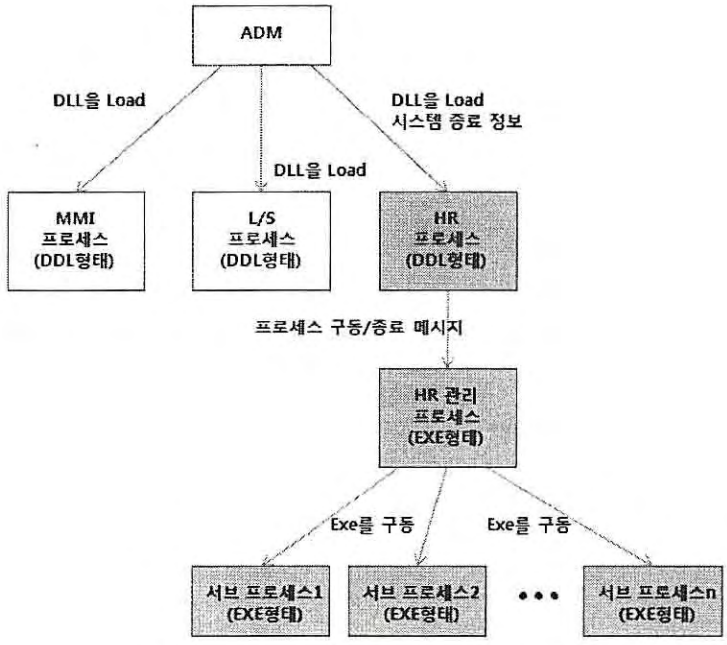


그림 3-23 홈 리스케줄링 ASCS연동

4. 수송관리 서브시스템

가. 개요

수송관리 서브시스템(SMS; Shunting Management System)은 현장역에 설치되어 운영되는 ASCS를 지원하는 서브시스템으로써 역의 수송원이 취급하는 입환 관리 등의 역 단위 열차운행과 관련된 업무를 시스템화하는 설비이다. 수송관리 시스템의 주목적은 역 단위의 열차 운행계획에 따라 역 구내에서 운용되는 입환 스케줄을 생성 및 변경하여 입환 지시서를 발행하고 타 서브시스템이 입환 스케줄이 결합된 온라인 스케줄을 사용하게 한다. 또한 현재 운행 중인 온라인 스케줄의 조회 및 본선스케줄 조회 기능도 수행한다.

나. 수송관리 서브시스템

(1) 기능 명세

(가) 온라인 스케줄 조회

현재 ASCS에서 사용하는 온라인 스케줄을 조회 하는 기능이다. 온라인 스케줄을 수송 관리자가 처음 실행 시 또는 지정된 스케줄 결합 시각이 입환 스케줄 과 본선스케줄을 결합해서 만들어진다. 이렇게 생성된 스케줄의 관리는 TMS-GUI가 담당하지만 역 구내에서 운용되는 입환 스케줄의 원활한 작성을 위해서 수송 관리자에서도 온라인 스케줄을 조회 할 수 있는 기능이 제공된다.

온라인 스케줄을 다음의 요소로 구성된다.

- 운행날짜
- 순번
- 열차 식별자
- 방향
- 도착/출발 진로
- 도착/출발 시각
- 열차구분
- 열차종별
- 시발역

- 종착역
- 실적출발/실적도착 시각
- 실적출발/실적도착 진로

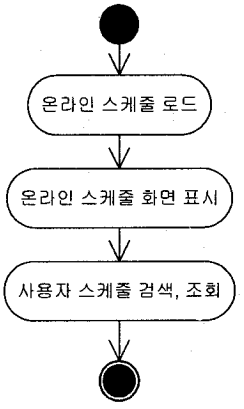


그림 3-24 온라인 스케줄 조회 흐름도

(나) 입환 스케줄 관리

역 구내에서 사용되는 입환 스케줄을 생성하고 삭제 또는 변경, 복사하는 기능이다. 일 단위의 입환 지시서를 생성할 수 있다. 생성된 스케줄은 파일 단위로 관리가 되며 사용자는 입환 스케줄 관리 대화창을 통해서 날짜별로 조회 할 수 있고 검색하길 원하는 조건으로 검색 할 수 있다. 작성된 스케줄은 다른 날짜에서도 사용할 수 있도록 복사할 수 있다. 복사 시 새로 복사 되는 스케줄의 운용날짜는 사용자가 지정한 대상 날짜로 변경된다. 입환 스케줄은 다음의 요소로 구성되어 있다.

- 운행날짜
- 열차식별자
- 출발시각
- 출발진로
- 순서
- 연계열차(이전 열번)
- 연계열차(이후 열번)

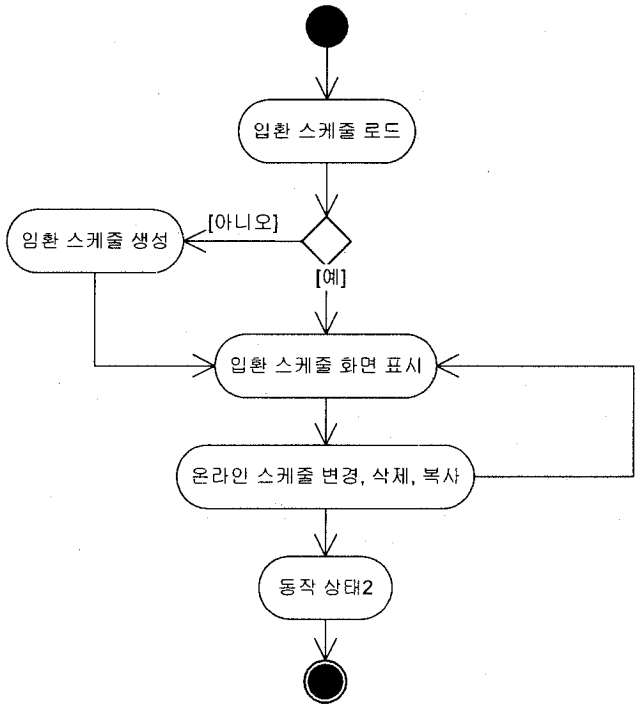


그림 3-25 입환 스케줄 관리 흐름도

① 입환 스케줄 생성

- 사용자는 입환 스케줄 작성 탭에서 해당 스케줄의 정보를 작성할 수 있다.
- 작성된 스케줄은 파일로 관리가 된다.
- 입환 스케줄 작성 시 열차 식별자의 입력 형식은 알파벳 한자리 + 숫자이다.
- 열차스케줄 작성 규칙에 어긋나는 정보를 입력하면 시스템은 해당 오류에 대한 정확한 메시지를 출력해서 잘못된 스케줄 정보가 입력되는 일을 방지한다.
- 입환 스케줄을 작성할 경우 자동 순서 부여 기능을 제공한다. 예를 들어 A1234 의 열차 식별자를 가진 스케줄을 작성할 경우 해당 열차 식별자와 동일하고 출발시각이 다른 스케줄을 추가로 입력한 경우 해당 스케줄들의 순서를 출발시각을 기준으로 자동으로 재 정렬하고, 연계열차의 앞, 뒤 열차로 자동으로 입력한다.
- 사용자는 또한 직접 입력 방법 외에도 마우스를 이용해서 연계열차 번호를 입력할 수 있다. 연계 열차 식별자 입력란을 선택한 후, 스케줄 테이블에서 입력하고자 하는 열차스케줄을 선택하면 해당 스케줄의 식별자(날짜+열차식별자+

순서)가 자동으로 입력된다.

② 입환 스케줄 변경

- 작성한 입환 스케줄을 선택 한 후에 사용자가 변경하길 원하는 스케줄 정보를 변경할 수 있다.
- 변경 작업이 완료 된 후, 입환 스케줄 관리 대화창을 닫을 시점에 시스템은 사용자에게 변경된 작업에 대한 저장 여부를 물어보는데, 이 때 사용자는 저장 여부를 선택할 수 있다.
- 사용자가 저장을 선택한 경우, 시스템은 변경 내용을 해당 스케줄 파일에 저장하게 된다.

③ 입환 스케줄 복사 삭제

- 이미 작성된 입환 스케줄에 대해서 시스템은 사용자가 해당 열차스케줄 별로 삭제 할 수 있는 기능을 제공한다.
- 또한 이미 작성된 일별 입환 스케줄에 대해서도 사용자는 일별 스케줄 전체를 삭제 할 수 있다.
- 사용자의 모든 삭제 작업은 입환 스케줄 관리 대화상자를 닫을 때 시스템이 사용자에게 변경 사항을 저장하겠냐고 물어 볼 때 저장을 선택 해야만 변경 사항이 적용된다.

(다) 오프라인 본선스케줄 조회

시스템이 LIS 또는 TMS-GUI 로부터 자율분산관리자에 의해서 전송된 본선스케줄 파일을 조회 할 수 있는 기능이다. 전송된 스케줄을 파일 형태로 관리되어진다. 시스템은 본선스케줄 파일을 전송 받아서 온라인 스케줄로 결합하는 데 사용하고 사용자는 시스템이 받은 스케줄을 조회 할 수 있다.

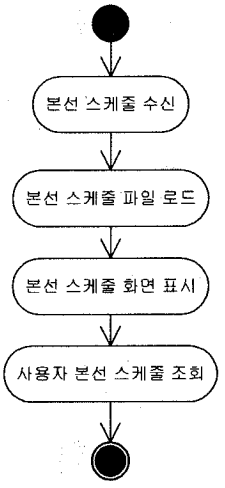


그림 3-26 본선스케줄 조회 흐름도

(라) 실적 스케줄 조회

시스템에서 지정한 날이 바뀌는 시각이 되면 오늘 기준으로 어제 스케줄은 삭제되고 오늘 스케줄은 어제 스케줄로 변하게 된다. 또한 오늘 기준으로 내일 스케줄이 오늘 스케줄이 되고 시스템은 내일 스케줄을 해당일의 본선스케줄 파일과 입환 스케줄을 사용해서 온라인 스케줄로 결합해서 사용하게 된다. 이렇게 ASCS에서 사용하는 온라인 스케줄에는 3일치의 스케줄이 존재하게 되는데, 이때 시스템은 운행이 끝난 스케줄은 실적 스케줄로 따로 관리한다. 실적 스케줄 조회 기능은 이렇게 만들어진 실적 스케줄을 사용자가 조회 할 수 있는 기능이다. 실적 스케줄은 파일로 관리된다.

(2) 구조

(가) SM 프로그램 패키지 구조

SM 프로그램은 스케줄 모델과 이 모델을 가공하는 컨트롤러, 가공된 스케줄을 화면에 표시하는 유저 인터페이스, 표시할 때 사용하는 유틸리티 패키지로 구성되어 있다.

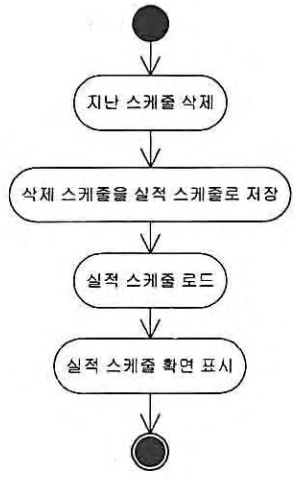


그림 3-27 실적 스케줄 조회 흐름도

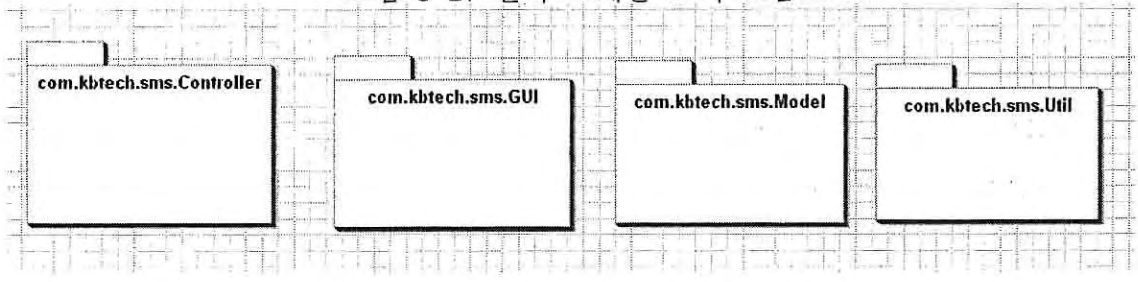


그림 3-28 SM 프로그램 패키지 구조

(나) SM 프로그램 클래스

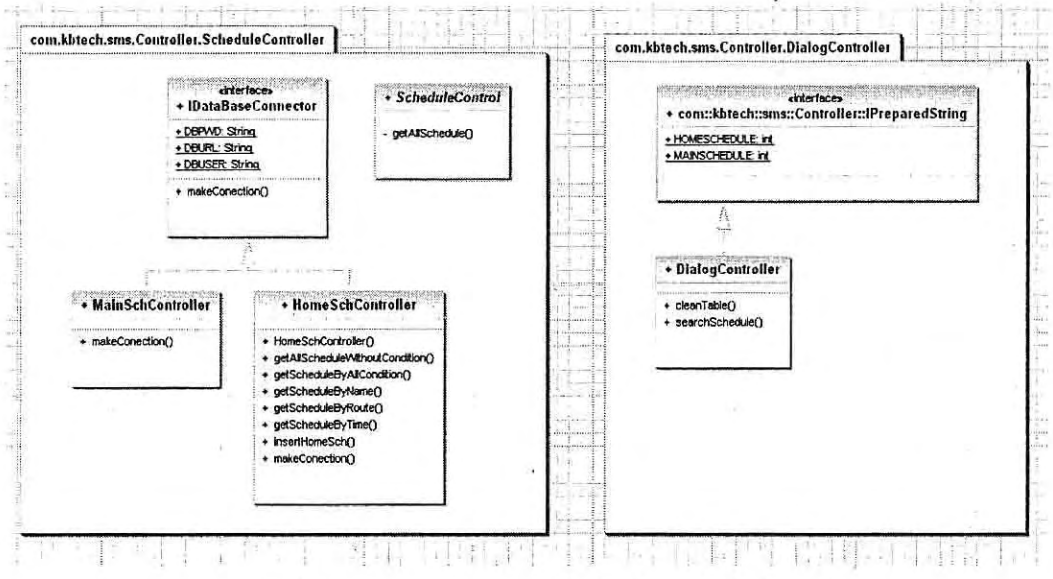


그림 3-29 SM 프로그램 클래스 다이어그램

(3) SM 프로그램 사용자 인터페이스

(가) SM 프로그램 전체 화면

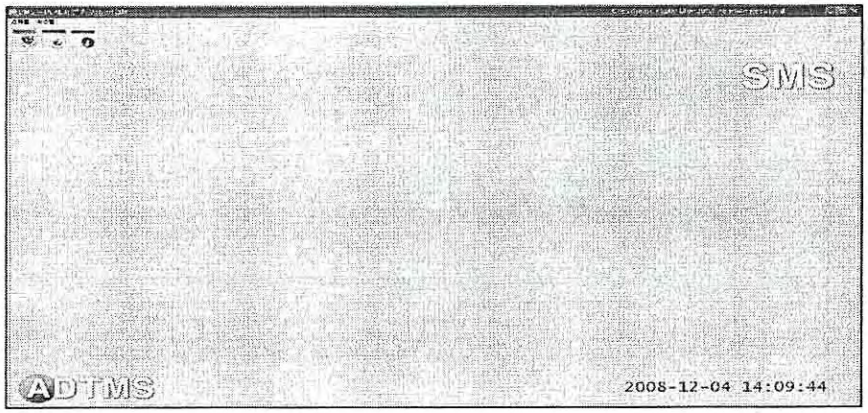


그림 3-30 SM 프로그램 전체 화면

SM 프로그램 전체 화면은 타이틀 바, 메뉴 바, 툴 바, 시계로 구성되어 있다.

(나) 온라인 스케줄 조회 대화 창

① 온라인 스케줄 전체 조회

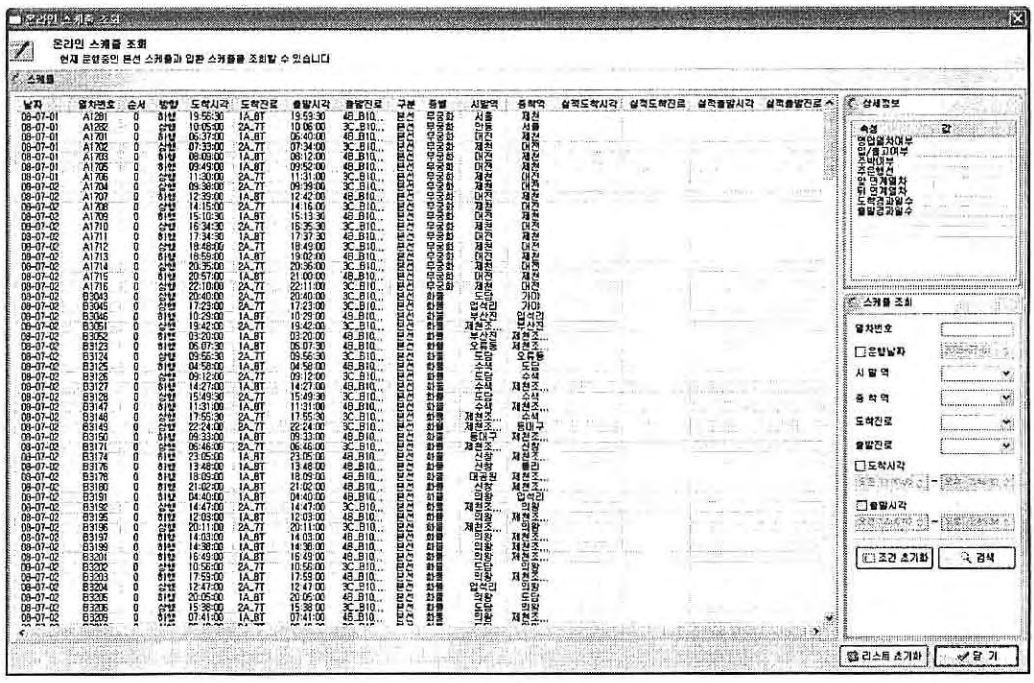


그림 3-31 온라인 스케줄 전체 조회 대화 창

(다) 입환 스케줄 관리 대화 창

① 입환 스케줄 날짜 선택



그림 3-32 입환 스케줄 날짜선택

② 입환 스케줄 로딩 대화 창

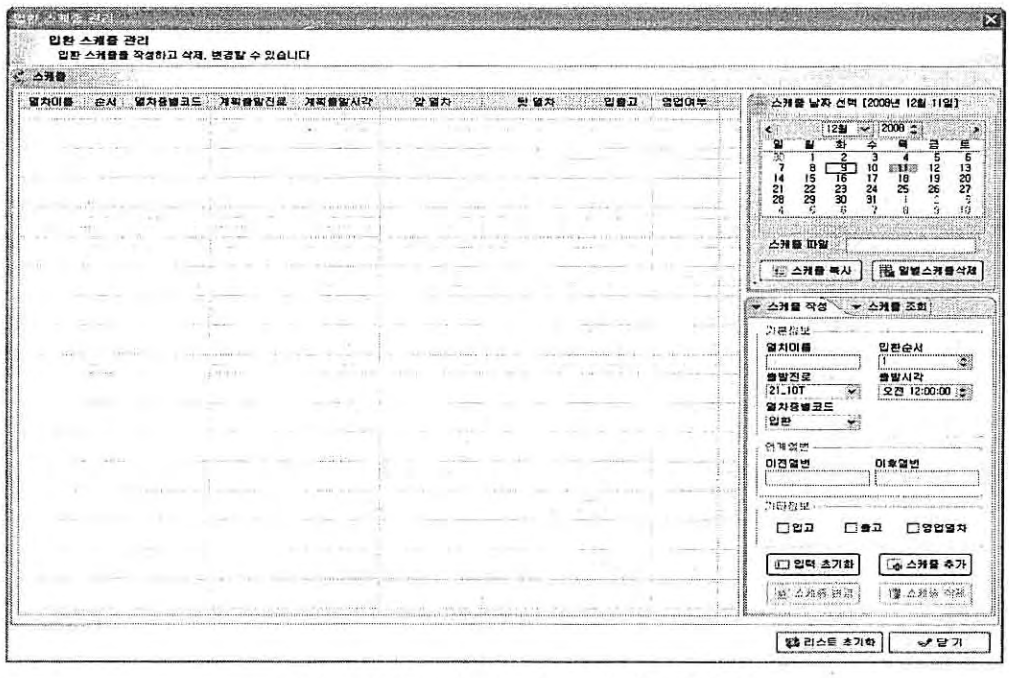


그림 3-33 입환 스케줄 관리 대화 창

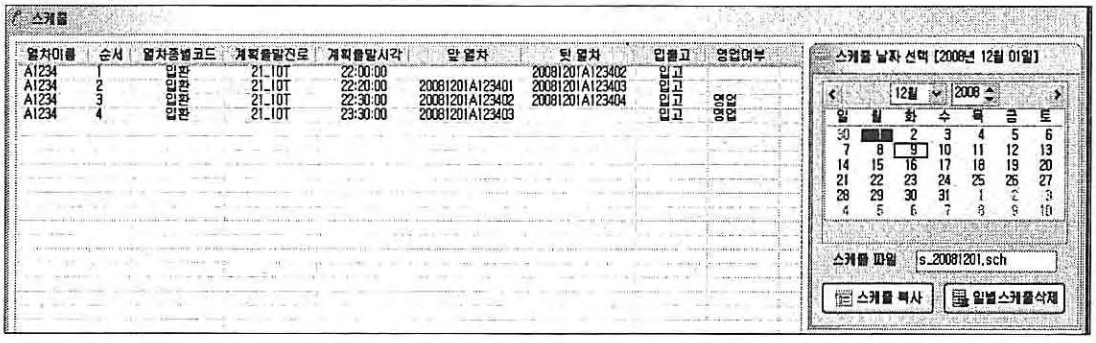


그림 3-34 입환 스케줄 표시

③ 일별 입환 스케줄 삭제



그림 3-35 스케줄 삭제

④ 입환 스케줄 새로 작성하기



그림 3-36 입환 스케줄 작성

⑤ 입환 스케줄 연계 열번 자동 입력

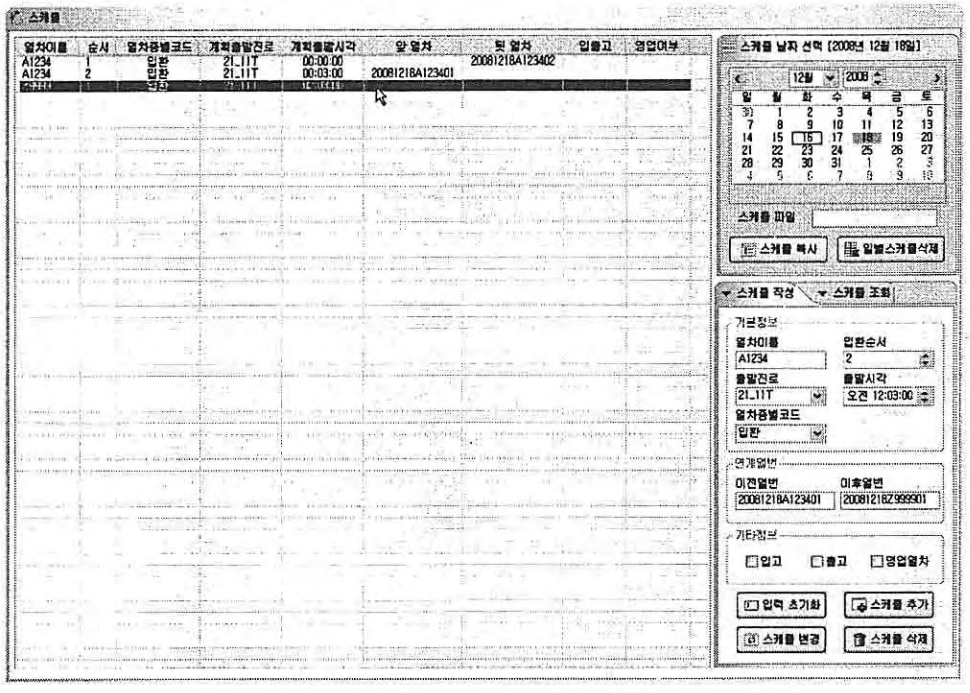


그림 3-37 입환 스케줄 연계 열번 자동 입력

⑥ 선택 입환 스케줄 삭제

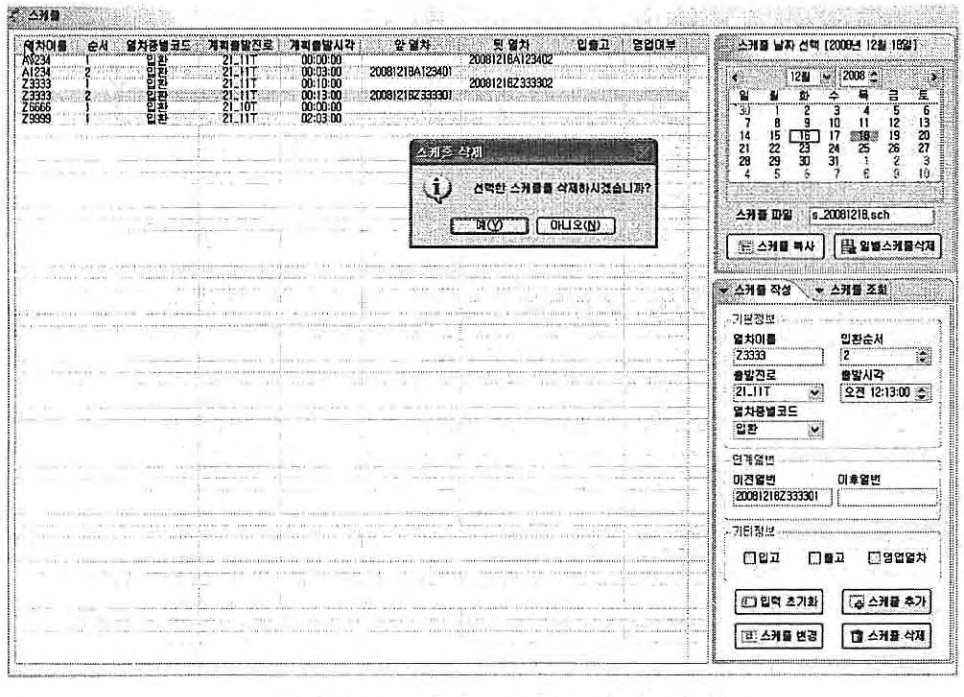


그림 3-38 입환 스케줄 삭제 확인

(라) 오프라인 본선스케줄 조회 대화 창

① 오프라인 본선스케줄 전체 조회 대화 창

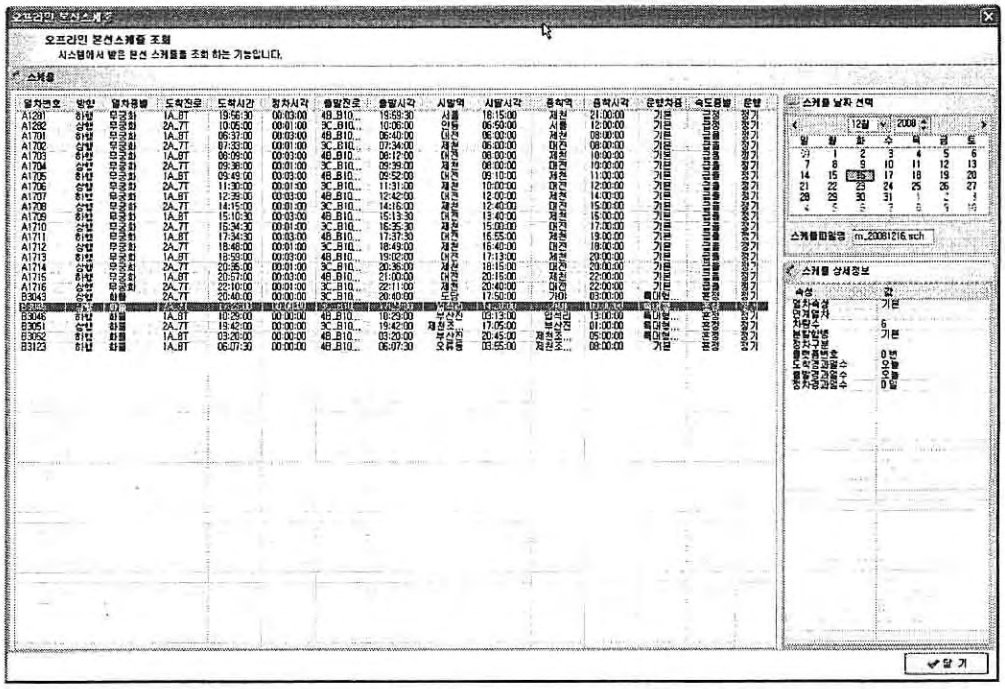


그림 3-39 오프라인 본선스케줄 전체 조회 대화 창

(마) 실적 스케줄 조회 대화 창

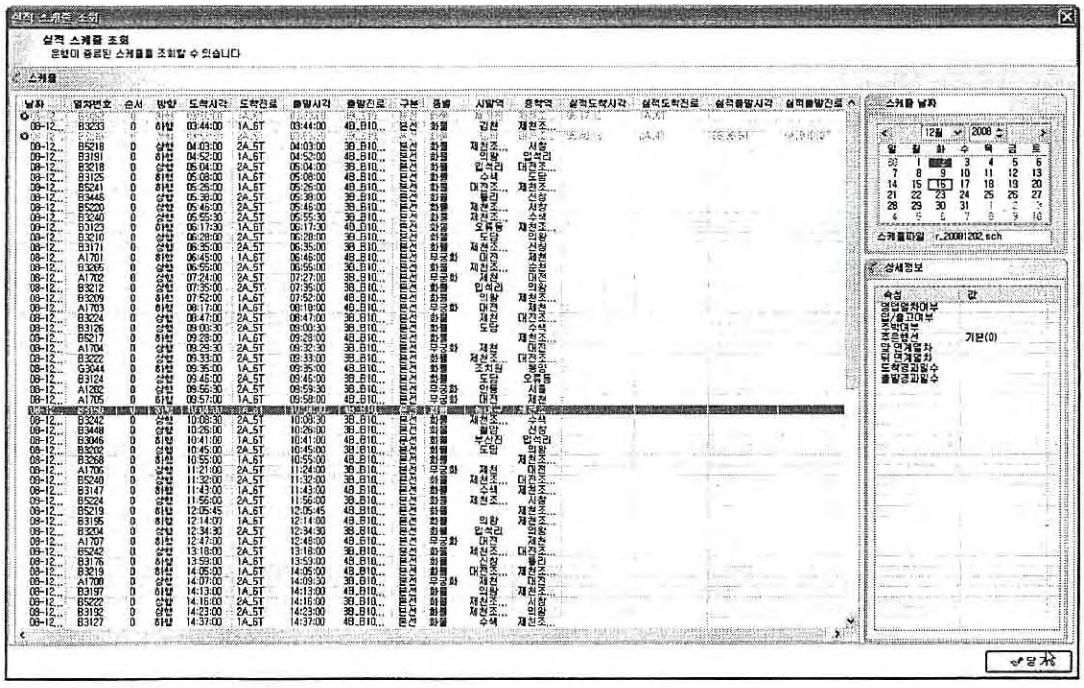


그림 3-40 실적 스케줄 조회 대화 창

5. 자율분산관리자

자율분산관리자는 하나의 서브시스템이 타 서브시스템의 영향 없이 자율성을 확보하여 자율제어를 할 수 있도록 하는 기능과 데이터필드를 통해 서브시스템간의 자율협조를 할 수 있도록 하는 주요 기능으로 구성되어 있다.

자율분산관리자는 5차 년도에 추가로 포함되는 여러 서브시스템에 대한 시험 결과 대량 데이터 및 여러 종류의 CC, 그리고 여러 개의 프로세스를 구동하는 경우 기능이 비효율적인 것으로 판단되어 설계 보완 및 시험을 수행하였다.

가. 핵심 용어 정의

(1) 데이터 필드

데이터 필드란 논리적인 네트워크를 의미하는 것으로 메시지를 송수신하는 가상의 채널이다.

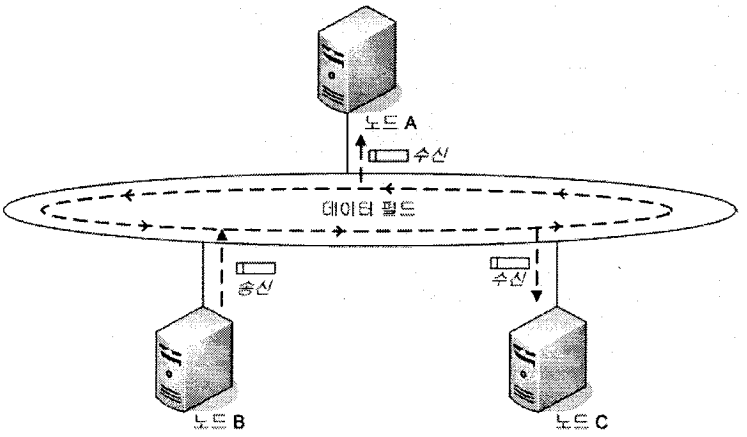


그림 3-41 데이터 필드간 송수신

네트워크에서는 메시지를 송수신할 때 메시지를 네트워크 영역에 뿌리는 기법 또는 해당 머신에 직접 접속하여 메시지를 전달하는 방식을 사용할 수 있는데 전자를 멀티 캐스트/브로드 캐스트 기법이라고 하고, 후자를 피어-투-피어 방식이라고 한다. 데이터 필드는 멀티 캐스트/브로드 캐스트 방식에서 사용하는 용어로써 자율분산관리자는 데이터 필드라는 가상의 네트워크 공간을 구성하게 된다. 멀티 캐스트/브로드 캐스트 방식으로 메시지를 송수신하기 위해서는 멀티 캐스트 그룹이라는 물리적인 그룹에 참여를 해야 한다. 이런 이유로 자율분산관리자는 가상의 네트워

크 영역인 데이터 필드를 구축하고 이 데이터 필드에 하나 이상의 멀티 캐스트 그룹을 구성한다.

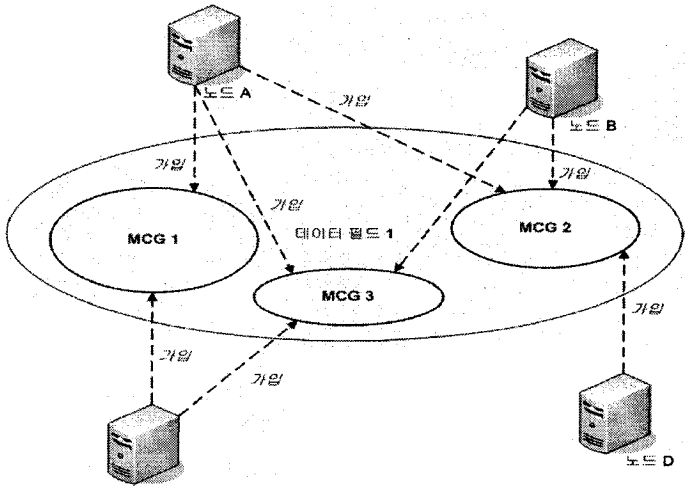


그림 3-42 멀티 캐스트 그룹

(2) 노드

자율분산관리자는 통신 미들웨어로써 다양한 송수신 방식을 제공해야 한다. 노드라는 개념은 개별 하드웨어에 고유하게 할당되는 번호로서 서로 다른 머신이 동일한 노드 번호를 가질 수 없으며 만약에 동일한 노드번호를 가진다고 할 때 메시지를 수신할 수 없게 된다. 개별 노드에 대해서 고유하게 노드 번호를 할당하면 메시지를 송수신할 수 있게 된다. 자율분산관리자는 메시지를 수신했을 때 동일한 노드에서 보낸 것인지 확인하게 되는데 이 과정은 멀티 캐스트 방식과 피어-투-피어 방식에게 모두 적용된다. 자신이 자신에게 메시지를 보내는 행위는 타당하지 못하기 때문에 자율분산관리자는 메시지의 헤더에 송신자의 노드 정보를 포함 하여 메시지를 송신하며 수신측 자율분산관리자는 이 정보를 토대로 동일 노드에서 송신한 것인지 확인하게 된다. 피어-투-피어의 경우 이 노드 정보는 아주 중요한 정보가 된다. 예를 들어 특정 메시지를 노드 A에게 송신하고자 하는 경우 자율분산관리자는 해당 메시지가 송신하려는 노드 번호에 대한 IP와 포트 번호를 알아내어 메시지를 송신하게 된다.

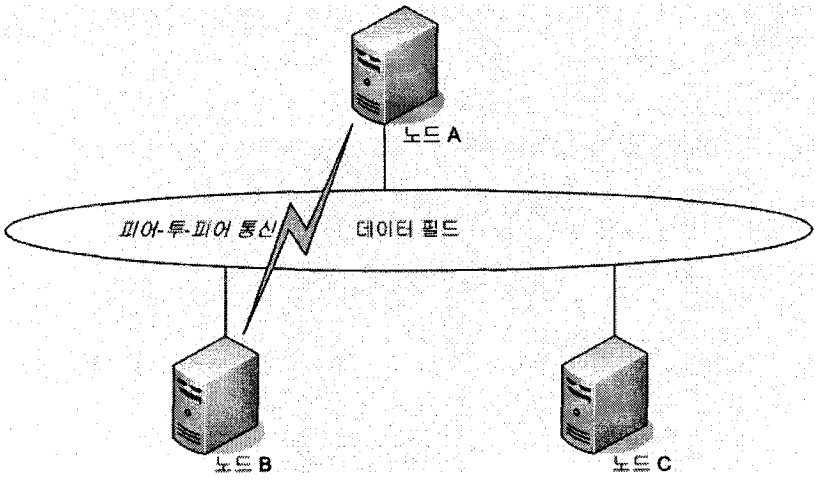


그림 3-43 노드간 피어-투-피어 송수신

(3) 내용 코드

자율분산관리자는 메시지를 송수신할 때 메시지를 선별하는 기준으로 내용 코드를 사용한다. 이 내용 코드는 의미 그대로 메시지의 내용에 대해서 코드 번호를 부여한 것을 말한다. 내용 코드는 해당 메시지에 대해서 고유하게 할당이 가능하며 내용 코드의 범위에 따라 다르게 사용된다.

내용 코드는 애플리케이션을 개발하는 개발자와 자율분산관리자의 유지보수를 담당하는 시스템 엔지니어에 따라 위와 같이 구분하여 사용할 수 있다.

자율분산관리자는 네트워크를 통해 메시지를 수신하면 메시지의 내용 코드를 통해서 수신여부를 결정하게 된다. 개별 애플리케이션은 자신이 수신할 수 있는 메시지가 정의되어 있고, 메시지 별로 내용 코드가 고유하게 할당되어 있기 때문에 자율분산관리자는 수신한 메시지를 사용할 수 있는 애플리케이션이 있는지 확인한 후 해당 메시지를 수신할 수 있는 애플리케이션에게 전달해 준다.

(4) 서브시스템

서브시스템이란 전체 시스템을 기능적으로 비슷한 기능을 제공하는 컴포넌트들로 묶고 그것을 하나의 독립적인 시스템으로 구분한 것으로써 자율분산시스템에서는 이 서브시스템을 다시 기능적으로 독립적인 프로세스라는 것으로 다시 구분하고 있다. 자율분산 역제어시스템을 구성하는 개별 서브시스템들은 그 역할에 따라 구

분되어 있으며 또한 서브시스템은 그 역할에 따라 프로세스로 구분되어 있다.

(5) 프로세스

자율분산관리자는 통신 미들웨어로써 메시지 송수신과 관련된 모든 기능을 제공하지만 실제 자율분산시스템을 구축하고자 할 때 자율분산관리자만 존재하는 경우 제대로 된 로직을 수행할 수 없다. 예를 들어 열차 추적의 경우 자율분산관리자는 열차 추적에 대한 기능을 제공하고 있지 않기 때문에 열차 추적과 관련된 기능을 제공하는 서브시스템을 구성하고 이 서브시스템 다시 열차 위치를 추적하기 위해서 필요한 다양한 기능을 독립적인 프로세스로 다시 구축하게 된다. 따라서 프로세스는 자율분산관리자를 통해 메시지를 송수신하게 되고 자율분산관리자는 프로세스의 생명주기를 관리하게 된다.

프로세스의 모든 생명주기는 자율분산관리자가 관리하게 되지만 생명주기에 대한 구체적인 책임은 프로세스 구현자에게 있다. 자율분산관리자는 하나 이상의 프로세스를 동적으로 구동하여 메시지를 송수신하고 처리할 수 있는 환경을 제공하기 때문에 미들웨어라고 칭한다.

(6) 이중랜

자율분산관리자는 이중랜을 통해서 하나의 랜에 장애가 발생했을 경우 장애에 적극적으로 대처하기 위해서 다른 랜으로 스위칭을 할 수 있는 기능을 제공한다. 피어-투-피어 통신의 경우 현용계 LAN에서는 데이터를 송신하고 대기계는 백업용으로 사용한다. 장애가 발생하면 현용계, 대기계 랜을 자동으로 전환한다. 멀티 캐스트 통신의 경우 현용계와 대기계 모두 송수신을 하지만 선착 데이터를 사용하고 후착 데이터는 폐기하는 정책을 사용한다.

나. 자율분산관리자

(1) 유즈케이스

(가) DF수신

DF의 데이터를 IDF(Internal Data Field)로 저장하며 다른 응용프로그램(APP,

Application Program)로부터 수신된 데이터도 ADM의 IDF에 저장한다.

(나) DF송신

ADM에 연결되어 있는 응용프로그램으로부터 데이터를 인계받아 테스트출력 처리한 후 IDF에 저장하고 DF로 데이터를 전송한다.

(다) APP구동

IDF에 저장된 데이터(Contents)를 이용하여 결함 진단 및 결함 마스킹을 하며 APP에게 구동 이벤트를 전달하고 테스트모드 및 테스트 데이터를 처리한 후 IDF의 정보를 삭제한다.

(라) APP 초기화

사용자가 ADM을 기동시키면 ADM은 기동시 다수의 응용프로그램을 로딩 해야 한다. 로딩 되는 응용프로그램은 초기화 루틴을 시작하며 동적으로 응용프로그램 자신의 CC, IDF, 구동정보를 ADM에 등록 한다.

(마) 온라인모드 설정

보수자가 ADM의 보조 GUI를 이용하여 APP 단위의 온라인/테스트 모드를 설정한다.

(바) APP 기동/종료

보수자가 ADM의 보조 GUI를 이용하여 APP 단위의 기동, 재기동 및 종료 명령을 수행한다.

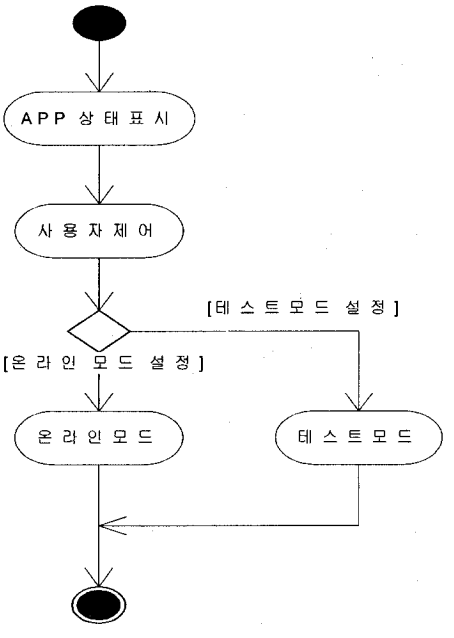


그림 3-44 APP제어

(2) 유즈케이스 명세

(가) APP모드 제어 유즈케이스

① 사전조건

ADM 및 응용프로그램이 정상적으로 동작하여야 하며 ADM 인터페이스 프로그램이 기동되어야 한다.

② 사후조건

응용프로그램은 온라인모드에서 테스트모드로 또는 테스트 모드에서 온라인 모드

로 변경동작하게 된다.

③ 기본흐름

- APP 상태 표시

ADM 인터페이스 프로그램을 실행하여 ADM의 동작 상태를 표시한다.

ADM 인터페이스 프로그램은 시스템 내부 통신을 이용해 ADM에 접속한다.

- APP별 동작모드 설정

ADM의 동작 상태를 온라인 제어할 수 있다.

APP별로 온라인/테스트모드로 설정할 수 있다.

- 온라인 모드

온라인모드로 동작하면 온라인 수신된 데이터는 온라인으로 테스트 수신된 데이터는 테스트 데이터로 출력을 한다.

④ 대체흐름

- 테스트 모드

테스트모드로 동작하면 수신된 온라인, 테스트 데이터 모두 테스트 데이터로 출

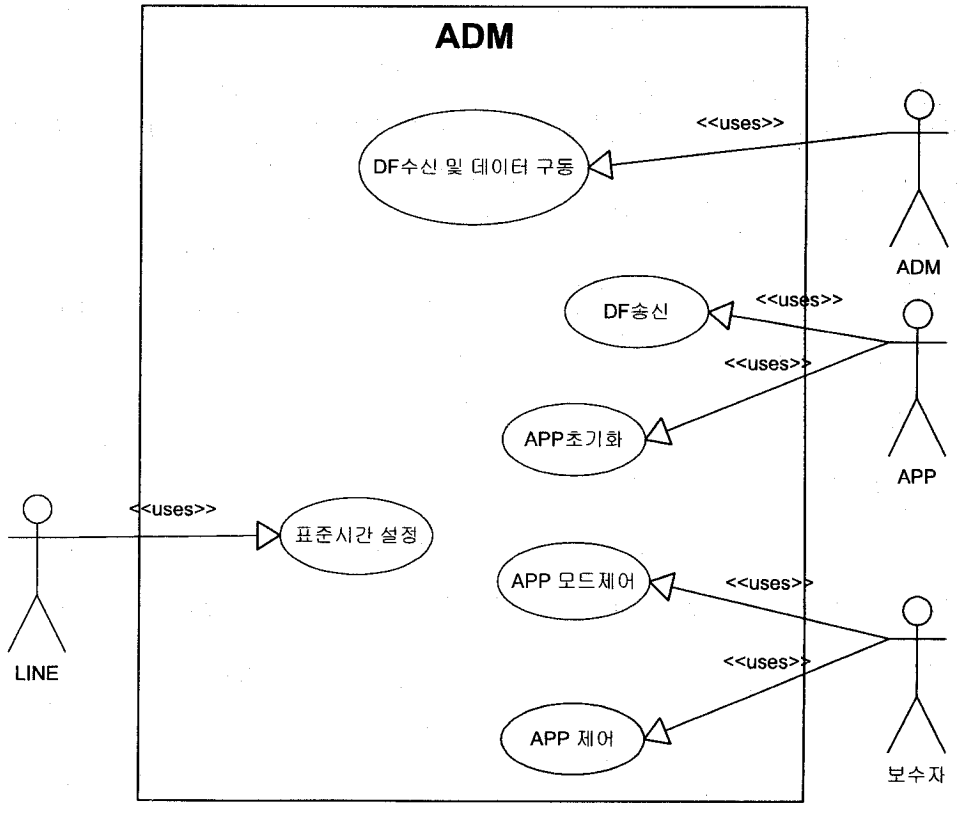


그림 3-45 ADM 유즈케이스

력된다.

(나) APP제어 유즈케이스

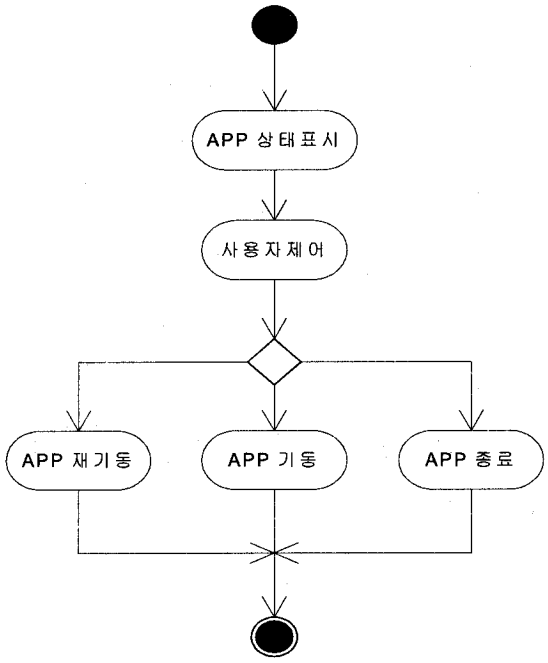


그림 3-46 APP제어

① 사전조건

ADM 및 응용프로그램이 정상적으로 동작하여야 하며 ADM 인터페이스 프로그램이 기동되어야 한다.

② 사후조건

응용프로그램이 다른 응용프로그램에 영향을 미치지 않으며 온라인상에서 재기동, 기동, 종료 된다.

③ 기본흐름

- APP상태표시

ADM 인터페이스 프로그램을 실행하여 ADM의 동작 상태를 표시한다.

ADM 인터페이스 프로그램은 시스템 내부 통신으로 ADM에 접속한다.

- APP별 사용자 제어

새로운 응용프로그램을 온라인상에서 기동한다.

동작중인 응용프로그램을 재기동 시킨다.

동작중인 응용프로그램을 종료 시킨다.

- APP기동

ADM에 응용프로그램이 추가 되는 경우 기동시킨다.

④ 대체흐름

- APP 재기동

동작중인 응용프로그램이 재기동 시켜야 하는 경우가 발생한 경우 사용된다.

- APP 종료

동작중인 응용프로그램이 특정한 이유로 가동중지를 시켜야 하는 경우 사용된다.

(다) DF송신 유즈케이스

① 사전조건 : 없음

② 사후조건

응용프로그램에서 생성된 데이터가 테스트 데이터 인지 확인 처리 후 데이터 필드로 송신되며 IDF에 복사된다.

③ 기본흐름

- 응용프로그램 처리종료

응용프로그램으로부터 처리 결과를 받는다.

서브시스템내의 출력을 기록하고 결함 진단 시 활용한다.

응용프로그램이 테스트 모드이거나 입력 데이터가 테스트 데이터 인지 확인하고 출력이 테스트 데이터인 경우 처리한다.

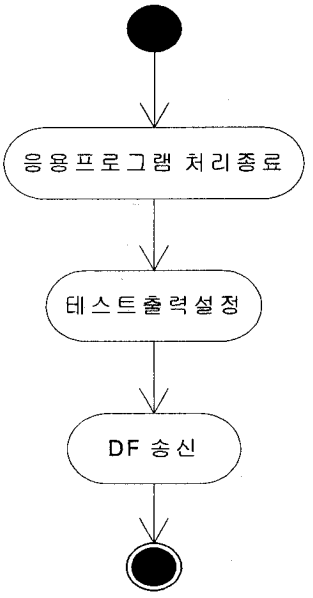


그림 3-47 DF 송신

- 테스트출력 설정

테스트출력으로 설정되어 있는지 확인하고 처리한다.(BIT)

- DF 송신

응용프로그램 APP구동 유즈케이스 실행중이면 대기후 IDF에 저장한다.

출력 데이터를 IDF 저장 후 DF에 송신한다.(멀티캐스트 방식 전송)

(라) DF수신 유즈케이스

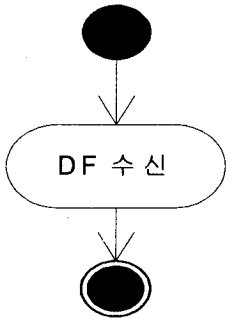


그림 3-48

DF수신

외부 서브시스템으로부터 발생된 데이터를 수신한다.

(마) 기본흐름

① DF 수신

- DF로부터 데이터를 수신한다.
- CC와 역번호로 필터링 된 데이터를 IDF에 저장.
- 응용프로그램 APP구동 유즈케이스 실행중이면 대기후 IDF에 저장한다.
-

(바) 대체흐름

(3) APP구동 유즈케이스

① 사전조건

- 처리를 위한 인터벌 타이머로 부터 처리 이벤트를 수신한다.
- 수신 관리자는 IDF에 데이터를 기록할 수 없다.

② 사후조건

- 응용프로그램이 구동하여 수신한 데이터를 처리하고 내부 데이터 필드는 삭제되며 응용프로그램의 출력이 발생된다.
- 다수의 응용프로그램에 모두 적용되어야 한다.

- 구동관리 처리 후 IDF를 삭제한다.

- DF수신 관리는 IDF에 데이터를 기록할 수 있다

③ 기본흐름

- 결함검지

중복된 응용프로그램으로부터 수신된 데이터를 비교하며 모두 같지 않으면 결함이 있다고 판단하고 마스킹 후 데이터를 사용한다.

다수결 동수인 경우 둘 다 결함으로 판단한다.(데이터 삭제)

- 응용프로그램 구동

응용프로그램 구동 조건을 조사하여 응용프로그램에게 메시지를 전달하고 다수의 응용프로그램에 모두 적용한다. 구동관리 종료 시 IDF를 삭제하며 테스트 데이터용 IDF도 동일한 처리를 한다. 응용프로그램이 온라인모드인 경우 테스트 데이터를 받으면 출력을 테스트로 설정한다.(BIT) 응용프로그램이 테스트모드인 경우 무조건(온라인, 테스트모드) 출력은 테스트로 설정한다.(BIT)

④ 대체흐름

- 결함진단

결함이 있는 수신 메시지(CC)가 서브시스템 응용프로그램의 출력인지 확인한다. 테스트 데이터는 처리하지 않는다.

- 결함제어

출력이 결함인 경우 DF 출력을 제한한다. Forward Recovery에 의해서 응용프로그램의 출력을 허용한다. 특별한 조건에 의해서 응용프로그램을 재기동 할 수 있다.

(4) 클래스 다이어그램

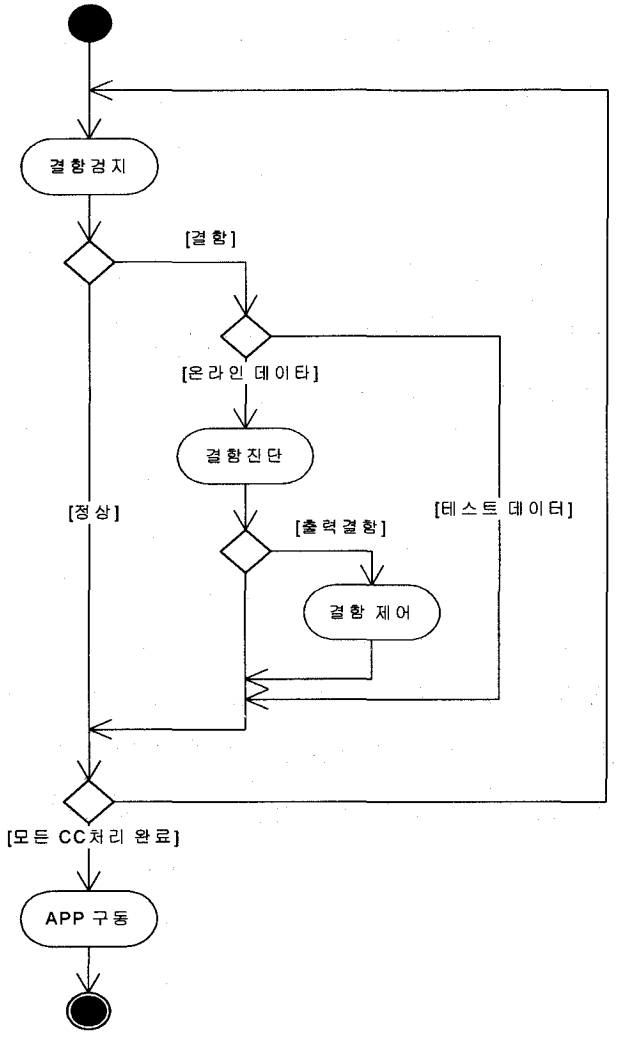


그림 3-49 APP 구동

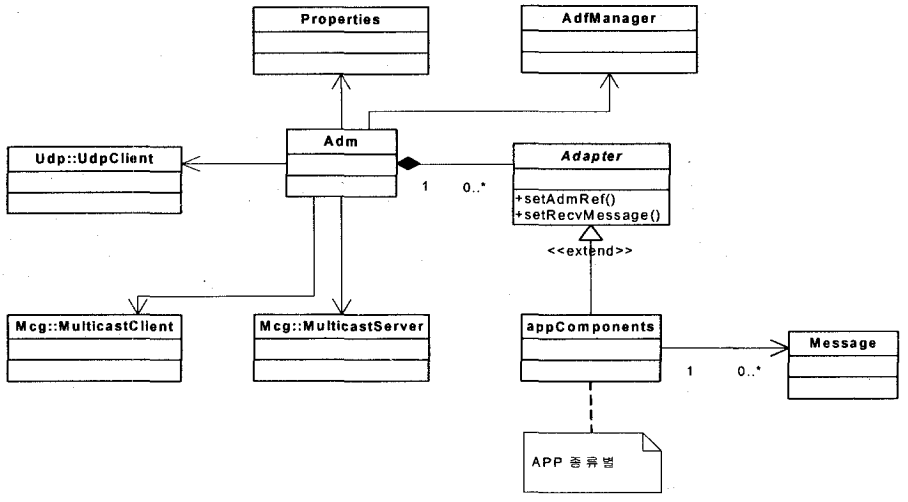


그림 3-50 ADM 클래스 다이어그램

(가) Adm

자율분산관리자이며 모든 리소스의 최종 관리를 한다.

main()함수를 가지고 있어 프로그램 시작점으로 사용된다.

버퍼링을 위한 송수신 큐가 있다.

응용프로그램을 로딩 한다.

기본 클래스들을 생성시킨다.

(나) Properties

응용프로그램의 초기 설정 파라미터를 로드한다.

(다) UdpClient

ADM의 실행시 사용자 커맨드를 수신하기 위한 Udp Client 통신을 한다.

(라) MulticastClient

데이터 필드로부터 콘텐츠를 가져온다.

(마) MulticastServer

콘텐츠를 데이터 필드에 전송한다.

(바) Adapter

다수 응용프로그램을 위한 인터페이스

응용프로그램과의 인터페이스를 담당한다.

응용프로그램이 구현한다.

(사) AdfManager

데이터 필드의 콘텐츠를 관리한다.

중복된 콘텐츠를 마스킹 한다.

쓰레드로 동작한다.

(아) Message

응용프로그램 간 사용하는 콘텐츠의 기본 형식

메시지를 판독하는 방법을 제공한다.

- 메시지를 생성하는 방법을 제공한다.
- 메시지를 복사하는 방법을 제공한다.

(5) 클래스 명세

(가) UdpClient Class

① 속성

- adm : ADM 지시자
- logger : log4j 지시자
- socket : udp 통신 지시자

② 메서드

- UdpClient() : 생성자
- startApp() : 응용프로그램을 기동한다.
- stopApp() : 응용프로그램을 종료한다.

(나) MulticastClient Class

① 속성

- adm : ADM 지시자
- loop : 통신 쓰레드 루프변수
- mcgAdd : 멀티캐스트 클라이언트 어드레스

② 메서드

- MulticastClient : 생성자

온라인/테스트 라인 통신망 구성이 가능하며 2중계 망구성이 가능하다. 다중 데이터 필드 구성이 가능하다.

- run : 쓰레드

(다) MulticastServer Class

① 속성

adm : ADM 지시자

loop : 쓰레드 루프변수

mcgAdd : 멀티캐스트 서버 주소

② 메서드

- MulticastServer() : 생성자

데이터 필드에 콘텐츠를 생성시킨다. 온라인/테스트 라인 통신망 구성이 가능하다. 2중계 망구성이 가능하다. 다중 데이터 필드 구성이 가능하다.

- NotifyMsg() : 응용프로그램 구동 메시지 전달

- waitMsg() : 데이터 필드의 메시지를 대기한다.

- run() : 쓰레드

(라) Adapter Class

① 속성

running : 응용프로그램을 통제한다.

adm : 응용프로그램의 ADM 지시자

appId : 응용프로그램의 식별자를 저장한다.

appVersion : 응용프로그램의 갱신 버전 기록.

loadSeq : 응용프로그램의 기동 순서

subSystemId : 응용프로그램의 역번호

TestMode :응용프로그램의 기동 상태(온라인/테스라인)

TestMsgFlag : 수신 메시지의 테스트 여부

② 메소드

set/get AppId() : 응용프로그램의 식별자 속성함수.

set/get AppVersion() : 응용프로그램의 버전 속성함수.

set/get AscsId() : 역번호 속성함수

set/get LoadSeq() : 응용프로그램의 기동 순서 속성함수.

getMaxStation() : ADM의 최대 관리역 수

set/get SubSystemId() : 서브시스템의 식별자 속성함수

set/get TestMode() : 테스트 모드 식별 속성함수

notifyMsg() : 콘텐츠 수신 이벤트

notifyMsgFromLdtsComm() :LDTS 메시지 수신 이벤트

setAdmrRef() : ADM 지시자 설정

setSendMessage() : 콘텐츠 송신 설정

stopApplication() : 응용프로그램 종료

checkRecvMessage() : 콘텐츠 수신 확인

runOnlineAdjStMsg() : 인접역 온라인 콘텐츠 처리

runOnlineStMsg() :당역 온라인 콘텐츠 처리

runTestlineAdjStMsg() :인접역 테스트라인 콘텐츠 처리

runTestlineStMsg() : 당역 테스트라인 콘텐츠 처리

(마) AdfManager Class

① 속성

adm : ADM 지시자

logger : log4j 지시자

② 메소드

adfManager() :생성자

clearAdf() : 내부 데이터 필드를 초기화 한다.

maskingMessage() : 내부 데이터 필드를 마스킹 한다.

(바) Adm Class

① 속성

recvQueue: 수신 데이터 저장 큐

sendQueue : 송신 데이터 저장 큐

② 메서드

Adm() : 생성자

checkRevMessage() : 데이터 필드 검사

getAppNumber() :APP 버전 확인

loadApp() :응용프로그램 로딩

main() :ADM 시작함수

recvMessage() : 콘텐츠를 데이터 필드에서 수신

setSendMessage() :콘텐츠를 데이터 필드에 전송

loadMulticastSocket() : 멀티캐스트 통신시작

다. 통신 방식의 구분

자율협조 논리를 구현하기 위한 통신 방식은 아래의 4가지로 구분되며 아래의 통신방식은 자율협조 시스템 구축을 위한 요건을 충족시킨다.

- 멀티캐스트 통신
- Peer-to-Peer(P2P) 통신
- 응답 대기형 통신
- 멀티캐스트 응답 대기형 통신

(1) 멀티캐스트 통신

멀티캐스트 통신은 하나의 시스템에서 동일한 그룹으로 정의된 시스템(1 ~ n개)에게 데이터를 일괄적으로 전송하는 방식이다. 멀티캐스트는 이더넷 환경의 UDP 프로토콜을 근간으로 하며 송신 시스템은 수신자 그룹의 지정된 포트를 설정하여 데이터를 전송(브로드캐스트)하게 된다.

멀티캐스트 통신은 송신 시스템에서 수신 시스템이 전송한 데이터를 수신하였는지의 여부를 확인 또는 판단하지 않는다는 점을 고려하여 동일한 형식을 가진 데이터를 주기적으로 전송하게 된다. 또한 각 데이터 패킷들은 수신 시스템에 도착할 때 전송한 순서대로 데이터가 도착한다는 보장이 없으므로 데이터에 일련번호를 부여하여 수신측에서 가장 최신의 데이터 패킷을 사용할 수 있도록 한다.

(2) Peer-to-Peer(P2P) 통신

Peer-to-Peer 통신은 한 개의 시스템에서 다른 시스템을 지정하여 데이터를 송수신하는 경우에 사용한다. 이 방식은 안정된 데이터를 송신측에서 수신측에 전송할 수 있다는 장점이 있으므로 분실되어서는 안 되는 데이터의 송수신에 사용한다.

P2P통신은 일반적으로 TCP/IP방식을 사용하게 되며 UDP를 사용하여 P2P 방식을 구현할 수도 있다.

(3) 응답 대기형 통신

응답 대기형 통신은 전송된 데이터에 대한 응답을 요구하는 프로토콜이다. 전송 시스템은 브로드캐스트 혹은 멀티캐스트 방식으로 데이터를 전송하며 그 데이터에 대한 응답을 기다리는 방식을 취한다. 송신측 컴퓨터는 수신측 컴퓨터에서 응답이 오지 않는 경우에 지정된 횟수만큼 데이터 재송신을 함으로써 데이터의 응답 대기를 연장할 수 있다.

(4) 멀티캐스트 응답 대기형 통신

멀티캐스트 응답 대기형 통신은 응답 대기형 통신 방식과 멀티캐스트 통신을 동시에 사용하는 방식이다. 전송측 컴퓨터는 등록된 그룹에 브로드캐스트를 통해서 데이터를 전송하고 각 시스템들의 응답을 기다린다. 수신측 컴퓨터들은 수신된 메시지에 대해서 자신의 컴퓨터에서 처리해야 할 데이터인 경우에 그 처리 결과 값을 전송 측 컴퓨터에 내용코드를 포함하여 전송하게 된다. 전송 측 컴퓨터는 응답을 대기한 컴퓨터에서 응답 결과가 모두 수신되는 경우에 해당 메시지에 대한 처리가 완료된 것으로 하며 일정 시간 이내에 응답이 없는 시스템에 대해서는 재전송 방식을 이용하여 완전한 데이터 처리를 하도록 한다.

라. 자율분산 통신 프로토콜을 위한 데이터 필드의 형식 구분

각 데이터 필드의 형식 구분은 자율분산 시스템에 사용되는 통신 방식을 적용하는 기준을 설정하기 위한 것이다. 데이터 필드의 형식 구분 기준은 서브시스템간의 데이터 송수신에 전송하기 위해서는 데이터 사용 특성에 따라서 멀티캐스트 혹은 P2P 방식으로 구분 될 수도 있으며 응답이 필요한지의 여부에 따라서 응답 대기형 혹은 비 대기형으로 구분이 될 수 있다.

(1) 브로드캐스트 데이터 필드

브로드캐스트 데이터 필드는 자율분산 시스템의 기본이 되는 데이터 필드이다. 브로드캐스트 데이터필드는 분산된 실시간 열차 감시 및 제어, 시스템 상태 감시에

사용되며 주로 역 내부 시스템 간에 사용된다.

(2) 분산 데이터 필드

분산 데이터 필드는 브로트캐스트 데이터 필드에 기초를 두면서 대용량 스케줄과 같은 정보 전송에 사용된다. 분산 데이터 필드의 활용은 분산된 구성원간의 동작에 관계없이 데이터의 업데이트가 가능하게 된다.

(3) 집중 데이터 필드

집중 데이터 필드는 집중화된 관리를 위해 전송 시스템이 수신 시스템에 응답을 요청하는 방식을 사용한다. 집중 데이터 필드는 각 시스템 간에 P2P로 데이터를 송수신하게 되며 유지보수를 위한 서비스 정보 및 정형화하기 힘든 일반 정보의 송수신에 사용된다.

마. 네트워크 제어기술

자율분산 시스템은 여러 가지 방식의 데이터 필드가 존재하며 각각의 데이터 필드에 대한 데이터 신뢰성 및 보호를 위해서 네트워크 전송 제어기술이 필요하며 프로토콜 헤더에 이러한 기술을 보완할 수 있는 정보들을 포함하여야 한다.

시스템의 구현에 필요한 네트워크 제어기술을 아래와 같이 구분된다.

- 우선순위를 위한 스케줄
- 우선순위 데이터에 대한 전송 제어 방식
- 버퍼 제어 정책
- 버퍼 제어를 위한 워터마크 삽입
- 주기적 전송 기술

바. 프로토콜 상세

(1) 데이터 패킷의 구성

자율분산 시스템에서 사용되는 시스템간 송수신 메시지는 다음 표 3-11과 같이 구성된다.

표 3-11 시스템간 송수신 메시지

구분	설명
공통 데이터 헤더(64바이트)	자율분산 관리자를 위한 데이터 헤더
사용자 데이터	사용자 정의 데이터
CRC32	공통 데이터 헤더부터 사용자 데이터까지의 CRC32

(2) CRC32

CRC는 메시지 구조의 Header영역과 Data영역의 모든 바이트에서 수행된다. 이것은 특정 CRC 다항식을 이용해 Data를 modulo-two 분할한 이후의 나머지이며, Data는 연속된 바이트 대신, 연속된 비트로 취급된다. Most significant Bit는 첫 바이트의 High Bit이고, Least significant Bit는 마지막 바이트의 Low Bit이다. CRC의 생성은 전송되는 정보의 무결성을 검사하기 위해 사용되며, CRC의 전송순서는 Low Byte 에서 High Byte 순서로 이루어진다.

CRC의 생성 다항식은 $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$ 이다.

(3) 공통 데이터 헤더

공통 데이터 헤더는 총 64바이트로 구성되며 자율분산 시스템에서 각 시스템간의 데이터 송수신을 위한 데이터 필드에 공통으로 적용된다. 이 헤더정보는 도메인 필드, 데이터 필드, 내용코드, 논리번호 및 자율분산 관리자에 필요한 4가지의 프로토콜 및 네트워크 전송기술을 사용할 수 있는 전체 데이터를 포함한다.

표 3-12 공통 데이터 헤더

위치	길이	필드 명	설명
0	4	Header	ADTMS를 위한 프로토콜을 명시 'ADTP'
4	4	Message Length	헤더 길이(64 Byte) + 데이터 길이(n)
8	4	Source Address	전송 측 IP address
12	4	Destination Address	수신 측 IP address
16	4	Time Stamp	메시지 발생 시간
20	4	Message Sequence	메시지 일련 번호
24	4	Message control type	UDP Multicast, UDP Multicast Inquire,

				UDP Multicast N Inquire TCP send(peer send) TCP inquire(peer inquire) TCP_RPL(peer reply)
28	4	응답 요청	Source Address	Inquire 요청 측 address
32	4		Ctrl Block address	Inquire Control block address
36	4		ID sequence	Inquire ID sequence number
40	4		ID parameter	Inquire ID parameter
44	2	TCD		Transaction Code = 내용코드
46	2	Program Version		Program version
48	1	Packet Kind		Packet 종류(ACK request, Packet request, Packet response)
49	3	Spare 1		
52	4	Packet Sequence		Packet sequence number
56	2	Message Mode		ONLINE, TEST mode
58	1	Protocol Version		추후 확장될 프로토콜에 대비
59	1	Message Priority		메시지 우선순위
60	1	Current Block Number		현재의 block 번호
61	1	Total Block Number		총 block 번호
62	2	Segment block size		분리된 Block의 크기

(4) 사용자 데이터

(가) ASCS간의 열차 출발 데이터

열차가 역의 경계 구간을 통과할 때 출발하는 열차번호를 다음 도착역으로 전송한다. 메시지의 전송은 Multicast를 이용한다.

출발 데이터를 위한 데이터 포맷은 아래 표3-13과 같다.

표 3-13 출발데이터 포맷

위치	길이	필드 명	설명
0	5	Train Number	출발 열차 번호
5	6	Start Station	출발 역 번호

11	6	Next Station 1	다음 역 번호 1
17	6	Next Station 2	다음 역 번호 2
23	6	Last Station	종착 역 번호
29	8	Route	출발 진로명
37	6	Depart Time	출발 역 홈 출발 시각(년: 월: 일: 시: 분: 초)

(1) Start Station : 출발 역 번호(현재 역 번호)

(2) Next Station 1 : 다음 도착 역 번호

(3) Next Station 2 : 도착 역 다음의 도착 역 번호

(4) Last Station : 열차의 최종 종착 역

(5) Route : 열차가 홈에서 다음 역을 향해 출발하는 진로 명

(6) Depart Time : 출발 역 홈 출발 시각(YYMMDD:hhmmss)

(나) CTC 스케줄 정보(역별)

CTC에서 ASCS에 열차 스케줄을 전송할 때 사용한다. 메시지 전송은 P2P와 Multicast를 병행하여 사용한다. CTC는 P2P에서 Server로 동작하며 ASCS는 Client로 동작한다. 1개의 스케줄 정보는 최대 64개의 열차번호에 대한 정보 전송이 가능하다.

표 3-14 스케줄 데이터 포맷

위치	길이	필드 명	설명	
0	1	Num Repeat	메시지 내의 반복 횟수	
1	6	Station Number	스케줄이 수행되는 역 번호	
7	5	Train Number	열차번호	1개의 열차번호 관련 Packet (크기 : 62 Byte)
12	6	Start Station	열차 시작 역	
18	6	End Station	열차 종착 역	
24	8	Arrive Route 1	도착 진로 1	
32	8	Arrive Route 2	도착 진로 2	
40	6	Arrive Time	도착 시간	
46	8	Depart Route 1	출발 진로 1	

54	8	Depart Route 2	출발 진로 2	
62	6	Depart Time	출발 시간	
68	1	Station Pass	역 정차 구분	
69 ~				Packet 반복

(1) Num Repeat : 메시지에 포함된 열차의 수

(2) Station Number : 스케줄이 수행되는 역 번호

(3) Train Number : 열차 번호

(4) Start Station : 열차의 스케줄 시작 역

(5) End Station : 열차의 스케줄 종착 역

(6) Arrive Route 1, 2 : 열차의 홈 진입 진로(2개의 연속된 진로가 필요할 경우에는 2까지 사용)

(7) Arrive Time : 열차 홈 진입 예정 시각

(8) Depart Route 1, 2 : 열차의 홈 출발 진로(2개의 연속된 진로가 필요할 경우에는 2까지 사용)

(9) Depart Time : 열차 홈 출발 예정 시각

(10) Station Pass : 역에 정차하지 않을 경우에는 '1', 그 이외에는 '0'

(다) 현장 제어 정보(Information)

한 개의 역에서 발생되는 현장 제어 정보를 주기적으로 전송할 때에 사용한다. 한 개의 Packet에는 한 개 역 전체의 제어 정보가 포함이 되어 있다. Control Data 의format은 별도로 정의하여 사용한다.

표 3-15 제어데이터 포맷

Offset	Length	필드 명	설명
0	2	Version Number	Higher Version(1) + Lower Version(1)
2 ~ 1023	n	Control Data	Packet에 포함되는 메시지

(1) Version Number : Packet이 적용된 버전 정보(0.1 ~ 255.255)

(2) Control Data : Format된 현장 제어정보

(라) 현장 표시(상태) 정보(Indication Information)

한 개의 역에서 발생되는 현장 표시(상태) 정보를 주기적으로 전송할 때에 사용한다. 한 개의 Packet에는 한 개 역 전체의 표시정보가 포함이 된다. Indication Data의 format은 별도로 정의하여 사용한다.

표 3-16 현장 표시정보 포맷

Offset	Length	필드 명	설명
0	2	Version Number	Higher Version(1) + Lower Version(1)
2 ~ 1023	n	Control Data	Packet에 포함되는 메시지

(1) Version Number : Packet이 적용된 버전 정보(0.1 ~ 255.255)

(2) Indication Data : Format된 현장 표시정보

(마) 표준 시각 정보

표준시각 정보는 DF의 특정 시스템에서 주기적으로 현재의 시각을 네트워크에 브로드캐스트 하며 수신한 시스템에서는 현재의 시각과 비교하여 다른 경우에는 수신된 시각으로 업데이트 한다. 표준 시각의 브로드캐스트는 30분 간격으로 수행한다. 표준시각 정보는 BCD 코드를 사용한다.

표 3-17 표준시각 정보 포맷

Offset	Length	필드 명	설명
0	8	Date	년(4) + 월(2) + 일(2)
8	6	Time	Hour(2) + Minute(2) + Second(2)

6. 핵심서브시스템의 데이터베이스

가. Entity Relation Diagram

그림 3-51는 핵심서브시스템에서 필요로 하는 데이터베이스의 관계도이다. 각 열차데이터베이스와 신호기, 선로 등의 데이터베이스는 심벌 데이터베이스로 연결되

어있다.

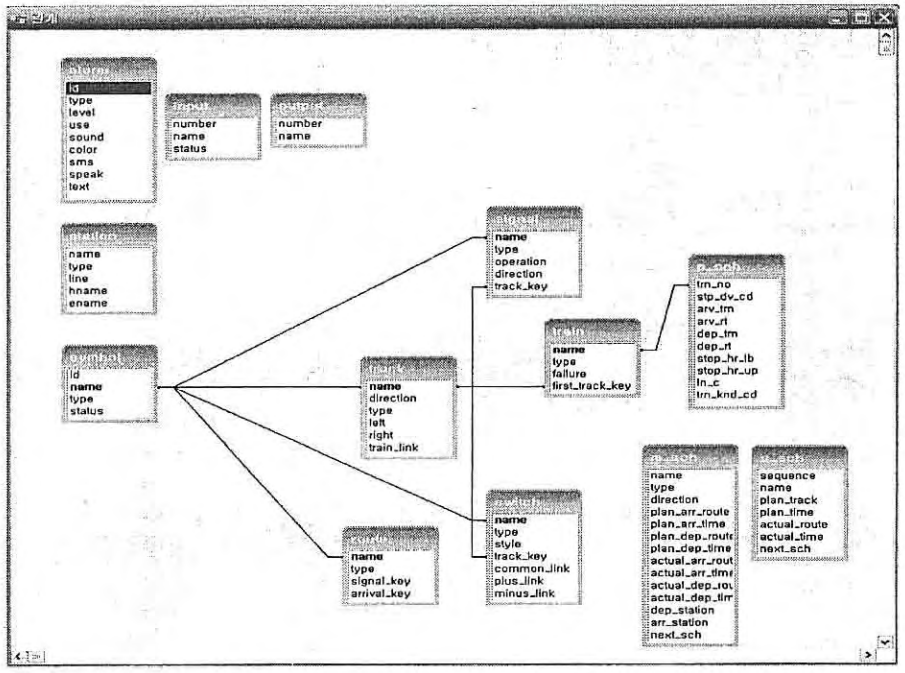


그림 3-51 ASCS-ERD

나. DB 테이블 명세

(1) 알람 테이블 (alarm)

알람 테이블은 ASCS에서 현장 상태정보의 변화 혹은 열차의 운행상황 감시 및 진로 제어 결과에 따라 발생되는 경보 및 이벤트 정보를 저장한다.

표 3-18 알람테이블

Field	Type	Key	Comment
id	int(10)	Pri	식별 번호
type	char(2)		시설물 타입
level	char(1)		메시지 수준(알람(경보), 이벤트, 취급)
use	char(3)		로깅 유무(E, D)
sound	char(3)		사운드 사용 유무(E, D)
color	char(10)		텍스트 색상
text	char(100)		문자열
grade	char(1)		경보등급(m:minor, M:Major, B:Block)

(2) 입력 테이블 (input)

입력 테이블은 전자연동장치와 관제시스템이 신호설비의 상태정보를 동일한 INDEX로 관리할 수 있도록 데이터를 저장한다. 신형 프로토콜을 적용하는 역에는 해당되지 않는다.

표 3-19 입력 테이블

Field	Type	Key	Comment
number	int(4)		비트별 입력 카드 인덱스(역별 초기화)
name	char(20)	Pri	입력 정보(계전기) 명칭
status	uint(4)		spare

(가) 필드 상세 설명

- number 필드는 역별로 0부터 n까지의 입력카드의 포인트 번호로 구성된다.
- name 필드는 역번호_타입_식별자로 구성된다.
- 상태는 동적으로 시스템에서 사용하는 것으로 입력하지 않는다.
- 공통은 1개의 타입으로 표시한다.
- 궤도는 1개 또는 2개의 타입으로 표시한다.
- 선호전환기는 4개의 타입으로 표시한다.
- 신호기는 1개 또는 다수의 정보로 필요에 따라 표시된다.
- 단선구간은 필요에 따라 다수의 정보로 표시된다.

(나) 공통 (Box Symbol)

- 020046_LOCAL (LOCAL MODE)->역별1개 존재
- 020046_CTC(CTC MODE)->역별1개 존재
- 020046_N1 (MAIN POWER N1)->역별1개 존재
- 020046_N2 (MAIN POWER N2)->역별1개 존재

- 020046_BC (BATTERY CHARGED)->역별1개 존재
- 020046_BD (BATTERY DISCHARGED)->역별1개 존재
- 020046_PS (CTC POWER SUPPLY)->역별1개 존재

(다) 궤도(구내+폐색)

- 020046_TK_201 (TRACK OCCUPY)-> 궤도별 존재
- 020046_WSP_201 (TRACK WSP)-> 진로 도착지별 존재

(라) 선로전환기

- 020046_SWP_21A (SWITCH PLUS)-> 선로전환기별 존재
- 020046_SWM_21A (SWITCH MINUS)-> 선로전환기별 존재
- 020046_SWL_21A (SWITCH LOCK)-> 선로전환기별 존재
- 020046_SWF_21A (SWITCH FAIL<IO>)-> 선로전환기별 존재

(마) 신호기(주신호+입환신호)

- 020046_SG_1A (SIGNAL)-> 신호기별 존재
- 020046_SGF_101 (SIGNAL FAIL<IO>)-> 사용되는 곳만 존재
- 020046_TTB_1A (SIGNAL TTB)-> TTB가 필요시 존재
- 020046_RI_1A (ROUTE INDICATOR)-> RI가 필요시 존재
- 020046_AL1_1A (APPROACH LOCK<ZAP1>)-> 신호기별 존재
- 020046_AL2_1A (APPROACH LOCK<ZAP2>)-> 신호기별 존재

(바) 단선신호(◁▷)

- 020046_DTF_201 (DOWN TRAFFIC DIRECTION)-> 필요시 존재
- 020046_UTF_203 (UP TRAFFIC DIRECTION)-> 필요시 존재

- 020046_DBR_201 (DOWN BRANCH RELAY)-> 필요시 존재
- 020046_UBR_203 (UP BRANCH RELAY)-> 필요시 존재

(3) 출력 테이블 (output)

출력 테이블은 연동장치에 제어 명령을 보내기 위해 제어 대상 신호설비의 INDEX를 정의한 것이다. 관제설비에 공통으로 적용된다.

표 3-20 출력 테이블

Field	Type	Key	Comment
number	int(4)		비트별 출력 카드 인덱스(역별 초기화)
name	char(20)	Pri	출력 정보(계전기) 명칭

(가) 필드 상세 설명

- number 필드는 역벌로 0부터 n까지의 입력카드의 포인트 번호로 구성된다.
- name 필드는 역번호_타입_식별자로 구성된다.
- 각포인트의 조합으로 제어를 메시지를 생성 한다.

(나) 공통

- 020046_LOCAL (LOCAL CONTROL)->역별1개 존재
- 020046_CTC(CTC CONTROL)->역별1개 존재
- 020046_CSSB (COMMON SIGNAL STOP BUTTON)->역별1개 존재
- 020046_ERRB (EMERGENCY ROUTE RELEASE BUTTON)->역별1개 존재
- 020046_TTB(TRAFFIC THROUGH BUTTON)->역별1개 존재
- 020046_TTCB (TRAFFIC THROUGH CANCEL BUTTTON)->역별1개 존재
- 020046_CPB (COMMON POINTER<SWITCH> BUTTON)->역별1개 존재

(다) 진로(주신호+입환 신호)

- 020046_SG_1A (SIGNAL PUSH BUTTON)->신호기별 존재
- 020046_TK_1T (ROUTE DESTINATION BUTTON)->도착지별 존재

(라) 선로전환기

- 020046_SWP_21 (SWITCH POSITION PLUS)->선로전환기별 존재
- 020046_SWM_21 SWITCH POSITION MINUS)->선로전환기별 존재

(4) 결합 스케줄 테이블 (schedule)

결합 스케줄은 ASCS에서 관리하고 있는 본선스케줄과 입환 스케줄을 하나의 테이블로 관리하기 위해 사용된다.

표 3-21 결합 스케줄 테이블

Field	Type	Key	Comment
name	char(5)		열차번호
sequence	int(4)		일련번호
trn_clsf_cd	char(2)		열차종별 코드
biz_trn_flg	char(1)		영업열차 여부 플래그
stck_dtw_flg	char(1)		입/출고 여부 플래그
jubak_flg	char(1)		주박여부 플래그
org_stn_cd	char(6)		출발역 코드
tmn_stn_cd	char(6)		종착역 코드
und_dv_cd	int(1)		상하행 구분
stop_dv_cd	char(2)		정차구분
mn_run_ln_cd	char(2)		주운행선 코드
plan_arv_tm	char(6)		계획도착 시간
plan_arv_elps_dno	int(1)		계획도착 경과일(dno)
plan_arv_rt_nm	char(20)		계획도착 진로
plan_dpt_tm	char(6)		계획출발 시간

plan_dpt_elps_dno	int(1)		계획출발 경과일(dno)
plan_dpt_rt_nm	char(20)		계획출발 진로
plan_stop_tm	char(6)		정차시간
prev_cnt_train	char(20)		이전연계 열차 sch_pk
next_cnt_train	char(20)		다음연계 열차 sch_pk
enable_flag	int(4)		스케줄 사용 표시
run_date	char(8)		스케줄 기준날짜(YYYY,MM,DD)
his_arr_time	char(6)		실적도착 시간
his_arr_route	char(20)		실적도착 진로
his_dpt_time	char(6)		실적출발 시간
his_dpt_route	char(20)		실적출발 진로
arr_route_set	int(4)		도착신로 설정 여부
dpt_route_set	int(4)		출발진로 설정 여부
sch_pk	char(20)	PK	날짜 + 열차번호 + 일련번호
hr_delete	char(1)		HR에서 사용하도록 정의

(5) 신호기 테이블 (signal)

표 3-22 신호기 테이블

Field	Type	Key	Comment
name	char(20)	Pri	신호기의 명칭(역번호+"SG"+식별자)
type	char(1)		주신호, 입환 신호(M, S)
operation	int(4)		현시(2, 3, 4, 5)
direction	char(1)		상행, 하행(R, L)
link	char(20)	Fn	연결 궤도 명칭(역번호+타입+식별자)
enter_track	char(20)	Fn	엔트리업 궤도

신호기 테이블은 연동도표에서 신호기의 특성 정보를 저장한다. 신호기 정보는 심볼 테이블에 정의되어 있어야 한다.

(6) 역 테이블 (station)

역 테이블은 ASCS역들의 역별 특성 정보를 저장한다.

표 3-23 역 테이블

Field	Type	Key	Comment
name	char(20)	PK	역번호
type	int(4)		종류
line	int(4)		노선
hname	char(20)		한글 역명
ename	char(20)		영문 역명
up_dir	char(1)		'L/S의 상행방향
cur_ascs	char(1)		현재 ASCS'
up_station0	char(20)		상행 연결역0
down_station0	char(20)		하행 연결역0
up_station1	char(20)		상행 연결역1
down_station1	char(20)		하행 연결역1

(7) 선로전환기 테이블 (switch)

선로전환기 테이블은 연동도표에서 선로전환기의 특성 정보를 저장한다. 선로전환기의 목록은 심벌 테이블에 정의되어 있어야 한다.

표 3-24 선로전환기 테이블

Field	Type	Key	Comment
name	char(20)	Pri	선로전환기의 명칭(역번호+"SW"+식별자)
type	char(1)		단동, 쌍동, 삼동(S, D, T)
direction	char(1)		UP(>-), DOWN(-<)(U, D)
track	char(20)	Fn	연결 궤도 명칭(역번호+타입+식별자)
common	char(20)	Fn	연결 명칭(역번호+타입+식별자)
plus	char(20)	Fn	연결 명칭(역번호+타입+식별자)
minus	char(20)	Fn	연결 명칭(역번호+타입+식별자)
LP	char(5)		spare
ref_switch1	char(20)		참조 선로전환기1
ref_switch2	char(20)		참조 선로전환기2

(8) 심볼 테이블 (symbol)

심볼 테이블은 연동장치와 LDTS, 그리고 CTC 서버에서 사용하는 현장 신호설비들에 대한 역 ELEMENT들을 순서대로 정의한 테이블이다.

표 3-25 심볼 테이블

Field	Type	Key	Comment
Id	Int(4)	Pri	심볼번호(엘리먼트 번호)
name	Char(20)	Pri	심볼의 명칭(역번호+타입+식별자)
type	Char(2)		심볼 타입(TK, SG, SW)
status	Int(4)		Spare (고장, 각 심볼의 상태값)
old_status	Int(4)		Spare (고장, 각 심볼의 상태값)

(9) 궤도회로 테이블 (track)

궤도회로 테이블은 심볼 테이블에 정의된 궤도회로에서 궤도회로별 연동도표 데이터를 기록한 것이다.

표 3-26 궤도회로 테이블

Field	Type	Key	Comment
name	char(20)	Pri	궤도의 명칭(역번호+"TK"+식별자)
type	char(5)		궤도 타입(도착, 출발, 장내, 폐색)
up	char(20)		하행 연결 궤도 명칭
down	char(20)		상행 연결 궤도 명칭
LP	char(5)		spare
platform	char(1)		Platform 궤도(1)/ 비 Platform궤도(0)
platform_type	char(1)		상(1)/하(2)/공통(3)
Platform_avail	char(1)		Platform 가용(1)/ 비가용(0)
siding	char(1)		사이딩 여부
siding_avail	char(1)		사이딩 궤도 가용 여부

(10) 진로 테이블(route)

진로 테이블은 연동도표에서 출발점이 신호기와 도착점인 궤도로 이름이 정의된 진로에 대해서 연동도표의 정보를 저장한다.

표 3-27 진로 테이블

Field	Type	Key	Comment
name	char(20)	Pri	진로의 명칭(역번호+"RT"+식별자)
ttb	char(1)		진로의 TTB 유무
switch	char(5)		진로내 선로전환기 방향
track	char(20)		진로내 궤도
signal	char(20)		진로 쇄정 신호기
Time_out1	char(5)		시소
Time_out1	char(1)		시소

(11) 열차 테이블 (train)

열차 테이블은 ASCS의 열차추적을 위해 사용하며 동적으로 열차의 목록이 관리된다. 열차 테이블의 업데이트는 RTPS에 담당한다.

표 3-28 열차 테이블

Field	Type	Key	Comment
name	char(5)	Pri	열차의 명칭(TR+열차번호)
type	int(4)		여객, 화물, 모타카…
status	int(4)		정상, 고장…
current	char(20)	Fn	열차 위치 궤도 명칭(역번호+타입+식별자)
dest	char(20)		진로 목적 궤도(역번호+타입+식별자)
dir	int(1)		열차 방향
date	char(8)		사용 스케줄의 run_date
seq	int(4)		입환 순서

3절 자율분산 노선관리시스템

자율분산 노선관리시스템은 노선 대상의 센터 시스템의 역할을 수행한다. 자율분산 노선관리시스템은 라인단위의 역에서의 열차 운행상황을 모니터링 하는 라인모니터링 서브시스템, 라인단위의 열차스케줄 조정이나 경합해소를 위한 라인경합관리 서브시스템, 스케줄관리를 수행하는 라인정보 서브시스템으로 구성된다.

1. 라인모니터링 서브시스템

가. 개요

라인모니터링 서브시스템은 라인 전체 혹은 지정된 역에 대한 신호설비 상태정보 및 열차 운행상황을 기호 형태로 운영자에게 표시하고 운영자가 본선 열차운행계획을 관리할 수 있도록 GUI를 지원하는 장치이다. 라인모니터링 서브시스템의 역 선형정보 화면은 그래픽 툴을 이용하도록 하며, 향후 역 선형정보가 수정되더라도 그래픽 툴을 이용한 역 화면 편집을 수행함으로써 프로그램의 변경 없이 새로운 역 선형정보를 표시할 수 있다.

나. 하드웨어 구성

라인모니터링 서브시스템은 1대의 컴퓨터 본체와 4대의 모니터, 키보드, 마우스로 구성된다. 첫 번째 화면은 열차 운행계획 관리를 위한 GUI로 이용되며, 나머지 3대의 모니터는 L/S 표시와 열차운행그래프를 표시하는데 사용된다. 라인모니터링 서브시스템의 구성도는 그림 3-52과 같다.

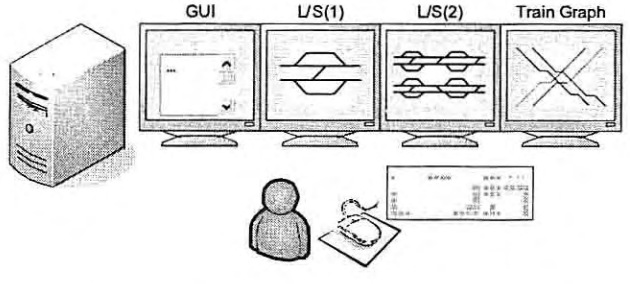


그림 3-52 라인모니터링 서브시스템의 구성

(1) 자율분산관리자

라인모니터링 서브시스템의 자율분산관리자는 ASCS의 자율분산관리자와 동일한 형식의 데이터를 송수신할 수 있는 구조로 되어 있으나 프로그램의 개발환경이 ASCS와 다르므로 동일한 인터페이스 구조를 가지는 또 하나의 자율분산관리자가 개발되었다.

라인모니터링의 자율분산관리자는 C++를 기반으로 되어 있으며, 자율분산 시스템의 특성을 준수하기 위하여 외부 데이터필드와 내부 데이터필드를 이용한다.

라인모니터링의 자율분산관리자를 위한 개발 소스 코드는 라인정보 서브시스템의 자율분산관리자 개발 소스 코드와 90%이상 동일하게 유지되고 있으며, O/S의 특성을 최대한 활용할 수 있도록 각 시스템에 따라 적용된 부분이 있다.

(가) 자율분산 시스템 환경을 위한 구성파일

① CCode 구성파일

```
; configuration for content's code:
;ccode    df      prio    mcg     response  chan    name            description
;         1       2       3       4         5       6       7       8
:12345678901234567890123456789012345678901234567890123456789012345678901234567890
1         1       1       1       0         0       cc1             ASCS 역표시
2         1       1       1       0         0       cc2             ASCS 역제어
3         1       1       1       0         0       cc3             ASCS 열차표시
4         1       1       1       0         0       cc4             ASCS 열차번호제어
5         1       1       1       0         0       cc5             표준시간
6         1       1       1       0         0       cc6             본선스케줄
7         1       1       1       0         0       cc7             시스템 모드 변경 수락 메시지
10        1       1       1       0         0       cc7             센터 열차번호
```

② 응용 프로그램과 ADM과의 IDF 구성파일

```
; Config for channel(mailbox) Connections
;
; chan : channel index
; type : P2P : P2P Communication, MCS : Multicast Send, MOR : Multiple Receive, SCD : Single-TCD Receive
; df   : id of datafield
; node : node number for p2p communication
; mcg  : Multicast group for destination in case MCS, SCD
; pri  : Priority
; flag : Send/Receive flag, 1=send, 2=rcv
```

```
; LAN  : Lan id, 1 or 2, 0 : single lan
; tmo  : Receive timeout(sec)
; intr : Interval at which a timer-transaction is posted(sec)
; ccode: Content's code to send, receive
#chn blk  type df   node mcg  pri  flag lan  tmo  ccode
;         1         2         3         4         5         6         7         8         9         10
;12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
1         MCS  1    0    1    1    2    1    0    1         1,10
2         MCS  1    0    1    1    2    1    0    1         1,10
```

③ P2P 구성을 위한 구성파일

```
; configuration for P2P Connection
; node : Node of server(client mode), Node of this system(server mode)
; node_addr1 : Node address
; node_addr2 : Node address(if dual lan is disabled, this field set to blank
; destport : port number to connect(in case server mode, destport is server port)
; sendport : port number to connect(in case client mode, send port is dedicated)
; retry    : retry to send data
; interval : interval(term) for message
;node     node addr1          node_addr2          destport  sendport  retry     interval
;         1         2         3         4         5         6         7         8         9         10
;12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
50        192.168.12.50       192.168.1.50        15000     25000     -1        10
2         192.168.12.47       192.168.1.47        15001     25001     -1        10
3         192.168.12.48       192.168.1.48        15002     2%002     -1        10
```

④ 데이터필드 정의

```
;datafield config file
#dfno     ip_mcast            ARPORT    ASPORT    ttl       dfname              description
;         1         2         3         4         5         6         7         8
;12345678901234567890123456789012345678901234567890123456789012345678901234567890
1         239.0.0.1           21000     21001     16        name1               data_field 1
2         239.0.0.2           31000     21001     64        name2               data_field 2 for sc
```

⑤ 멀티캐스트 그룹 정의

```
; Config for multicast group
;
;no       O_Port    T_Port    BUFSIZE   RETRY     NAME                nodelist
;         1         2         3         4         5         6         7
;123456789012345678901234567890123456789012345678901234567890123456789
1         20001     20002     512       1         mcg1                1,2,3     description 1
2         45000     45002     512       1         mcg2                          description 2
```

⑥ 시스템 특성 정의

```
; system information
:key        value
;           1          2
:12345678901234567890123456789
MODE        0
NAME        linecenter
OSTYPE      1
LINEID      1
NODEID      100
IP_ACT      192.168.12.210
IP_STBY     192.168.11.42 ;
```

(나) 자율분산관리자 구동 화면

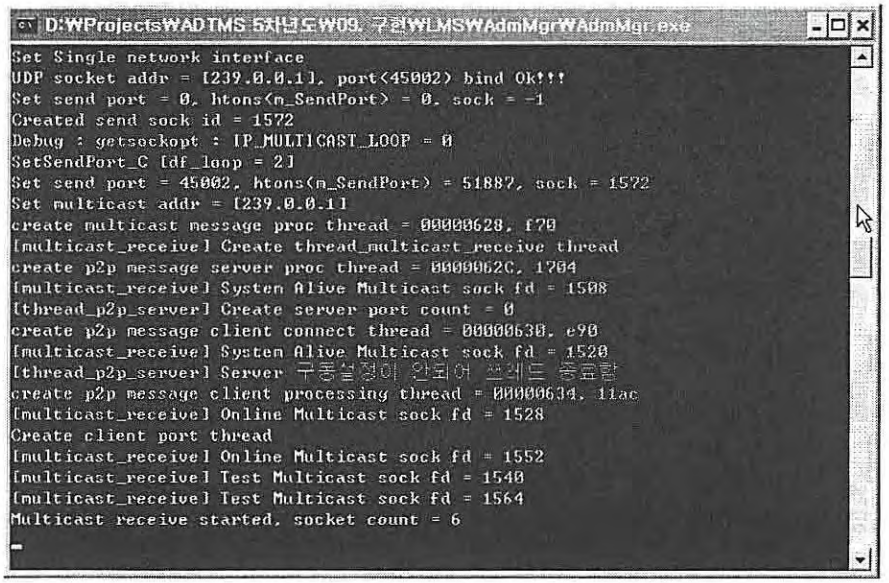


그림 3-53 라인모니터링의 자율분산관리자

(다) 테스트 분석을 위한 데이터 로그

시스템의 개발 및 시험을 위하여 동일한 오류를 발생시키거나 동일한 행위를 일으키는 데이터의 목록이 준비될 수 있다. 이러한 데이터는 논리적 접근에 의해서 순차적인 정보를 파일로 저장하고 이 정보를 데이터필드에 전송함으로써 시스템의 기능을 테스트할 수 있다. 라인모니터링의 자율분산관리자는 EXT에서 수행 가능한 데이터의 처리를 위하여 데이터필드에 송수신되는 모든 정보를 로그로 기록하도록 설정할 수 있게 설계되었다.

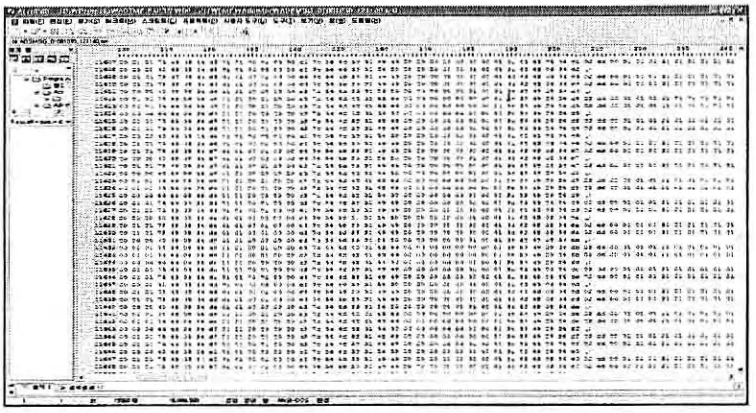


그림 3-54 자율분산관리자에 의해 저장된 데이터

(2) 라인모니터링 화면

라인모니터링 콘솔에 표시되는 화면의 구성은 초기 청량리, 성북, 망우, 중랑역을 대상으로 하여 구축되었으나 종합연계시험센터 및 현장 설치시험을 위한 준비 과정에서 대상 역을 충북선 오송, 청주, 오근장역으로 선정하고 이에 대한 화면 데이터 입력 및 시험을 수행하였다.

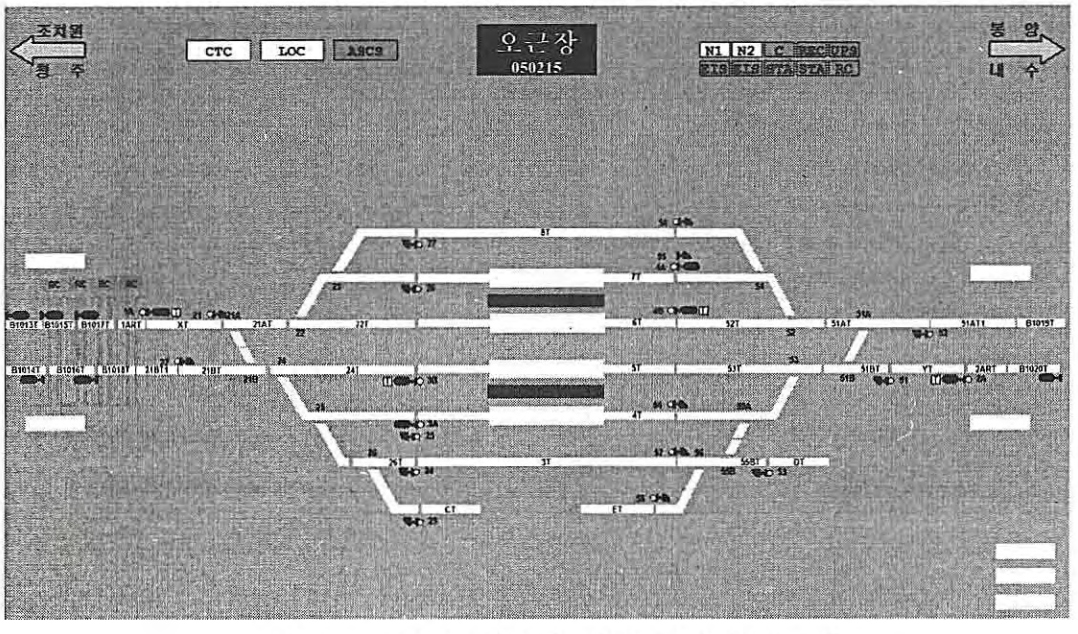


그림 3-55 라인모니터링 시험 대상역(오근장)

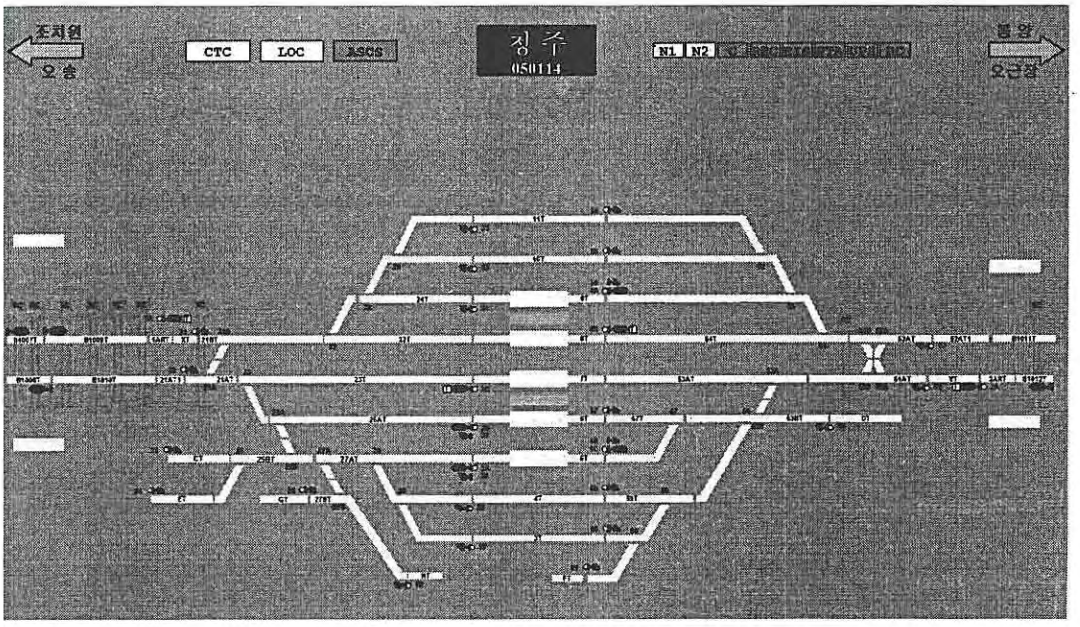


그림 3-56 라인모니터링 시험 대상역(청주)

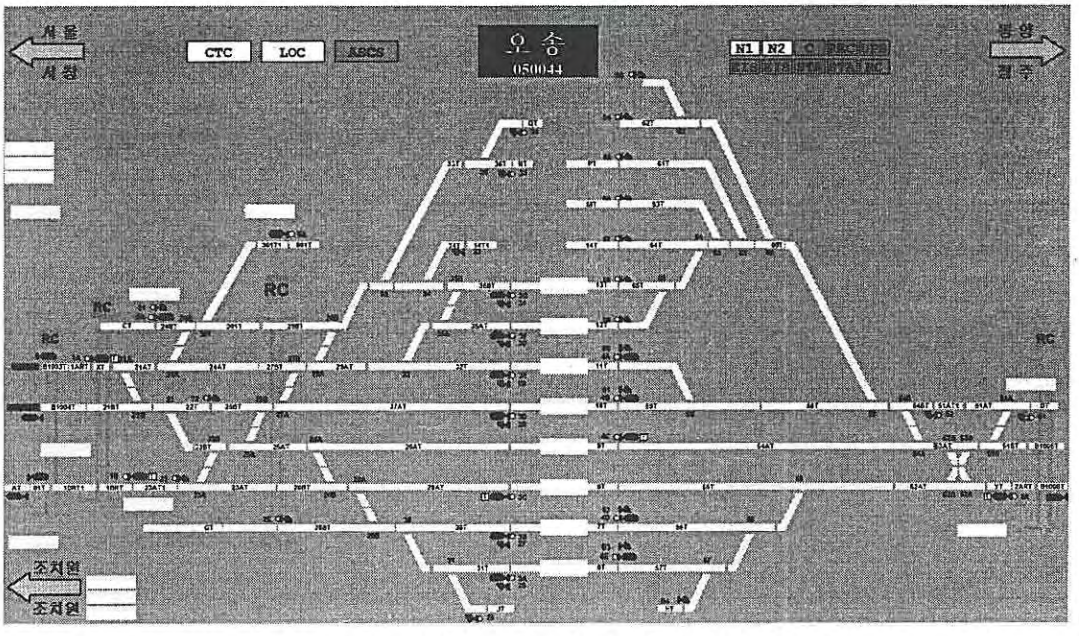


그림 3-57 라인모니터링 시험 대상역(오송)

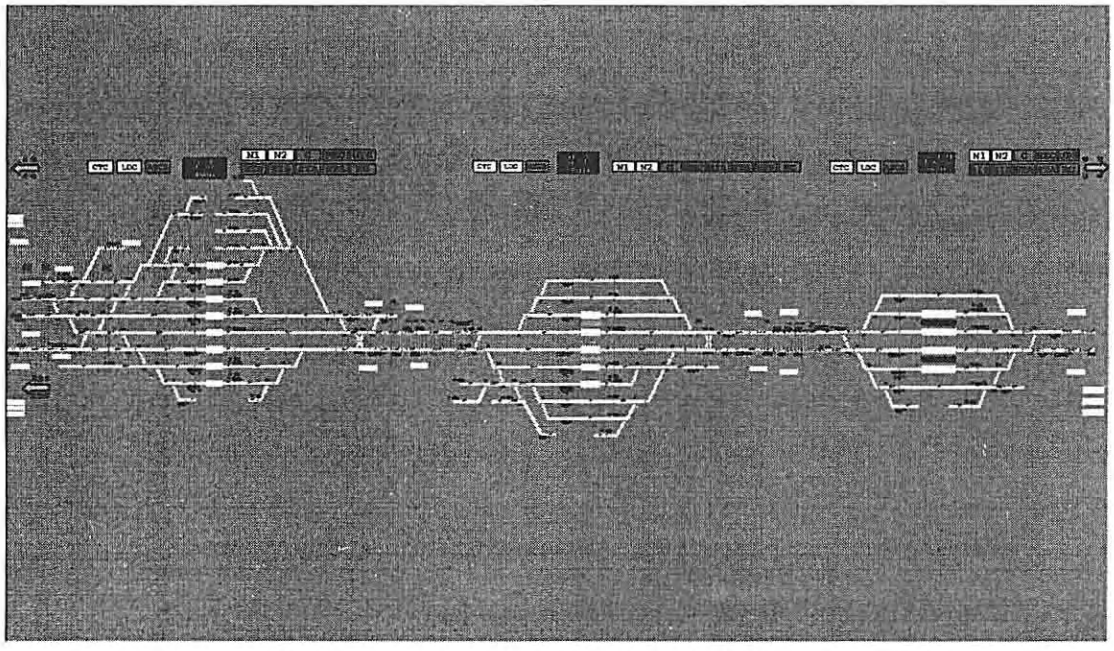


그림 3-58 라인모니터링 시험 대상 라인(충북선:오송-오근장 구간)

(가) 라인모니터링 표시를 위한 자료 흐름도

① 역 정보의 입력과정

Case 1 : 1 → 2 → 3 → 4

Case 2 : 3 → 4

② 역 정보의 변경 과정

Case 1 : 1 → 2 → 3 → 4

Case 2 : 5 → 2 → 3 → 4

③ 역 정보와 L/S-View의 동기화 과정

Case 1 : 1 → 6

Case 2 : 1 → 2 → 3 → 6

Case 3 : 5 → 6

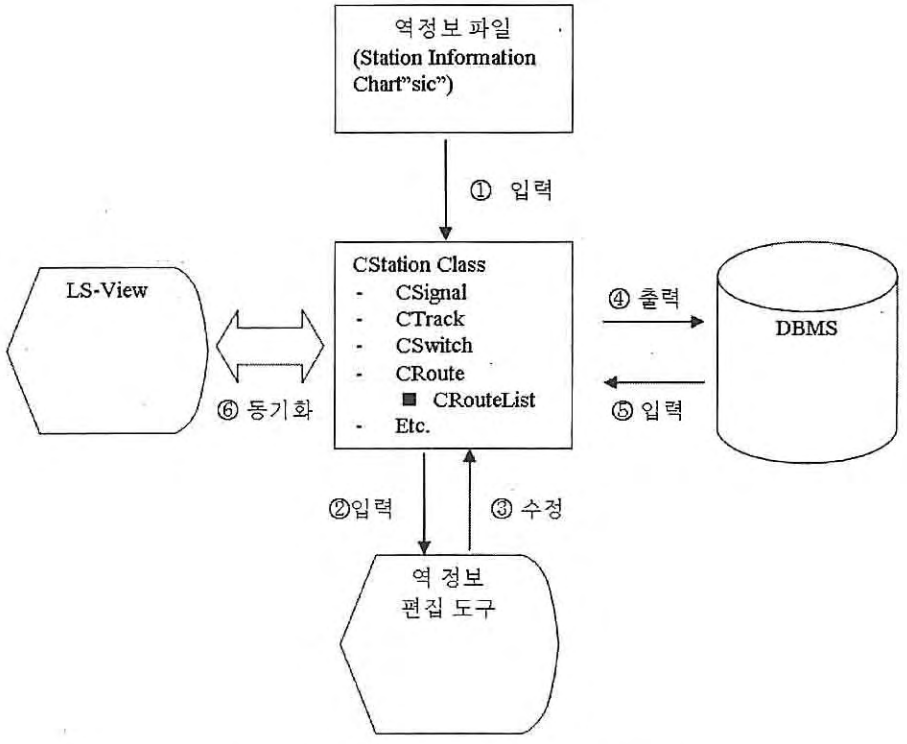


그림 3-59 라인모니터링 자료 흐름도

(나) 열차 운행계획 관리 화면

① 스케줄 관리 화면- 오프라인 열차스케줄 ID 관리

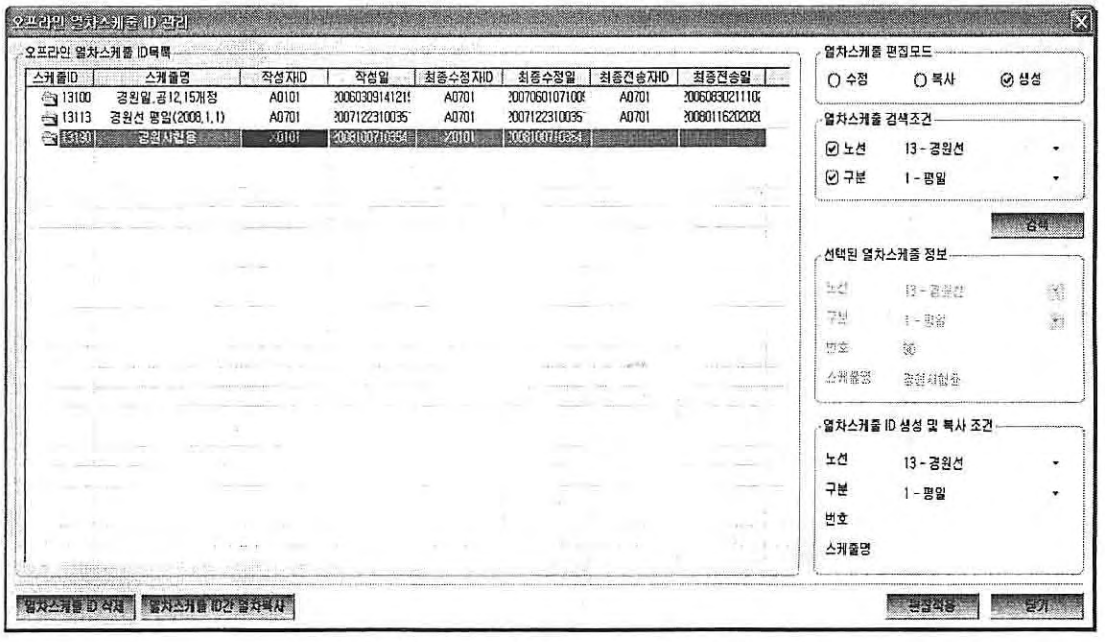


그림 3-60 오프라인 열차스케줄 ID 관리

② 스케줄 관리 화면 - 온라인 열차스케줄 ID 관리

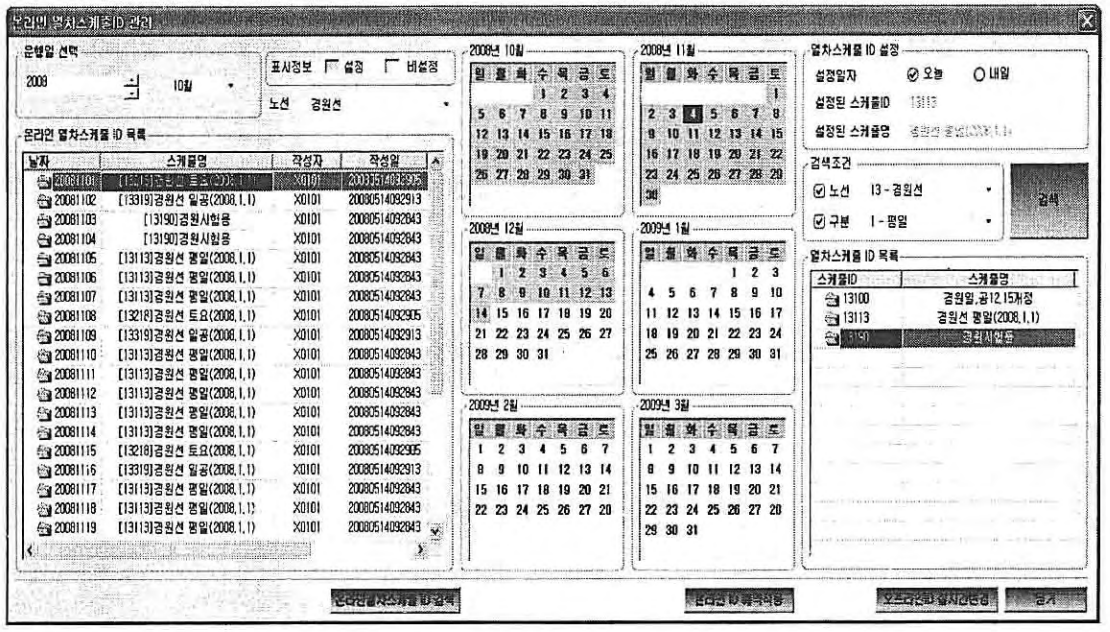


그림 3-61 온라인 열차스케줄 ID 관리

③ 스케줄 관리 화면 - 진입 예고열차 정리

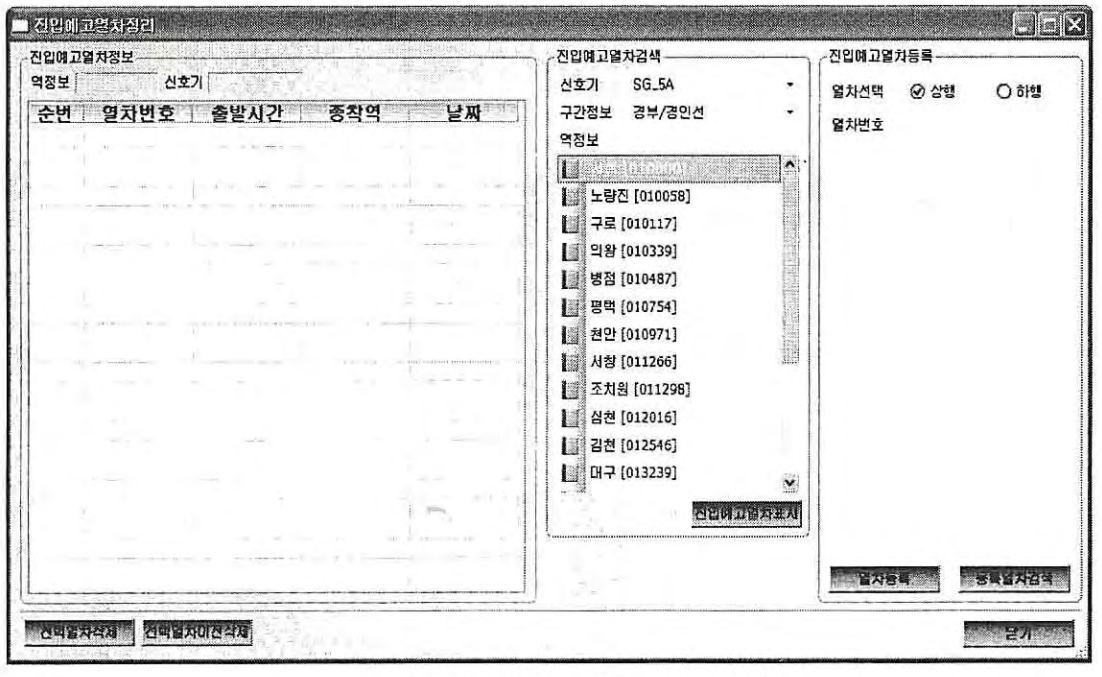


그림 3-62 진입 예고열차 정리

(다) 스케줄 관리 화면 - 오프라인 열차스케줄 관리

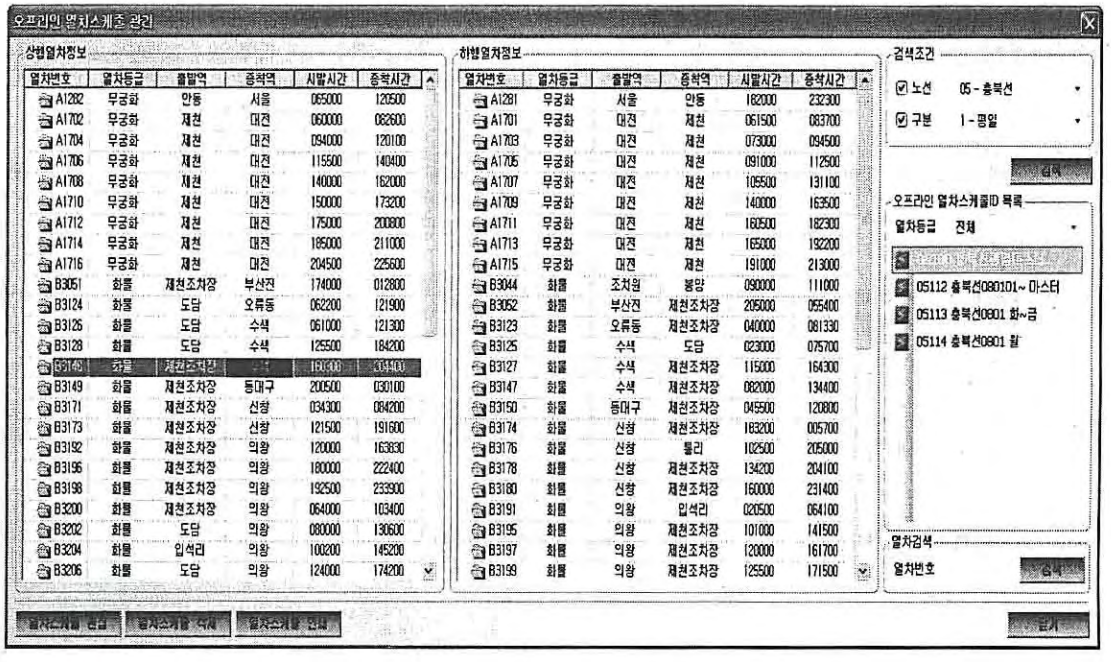


그림 3-63 오프라인 열차스케줄 관리

(라) 스케줄 관리 화면 - 오프라인 열차스케줄 편집

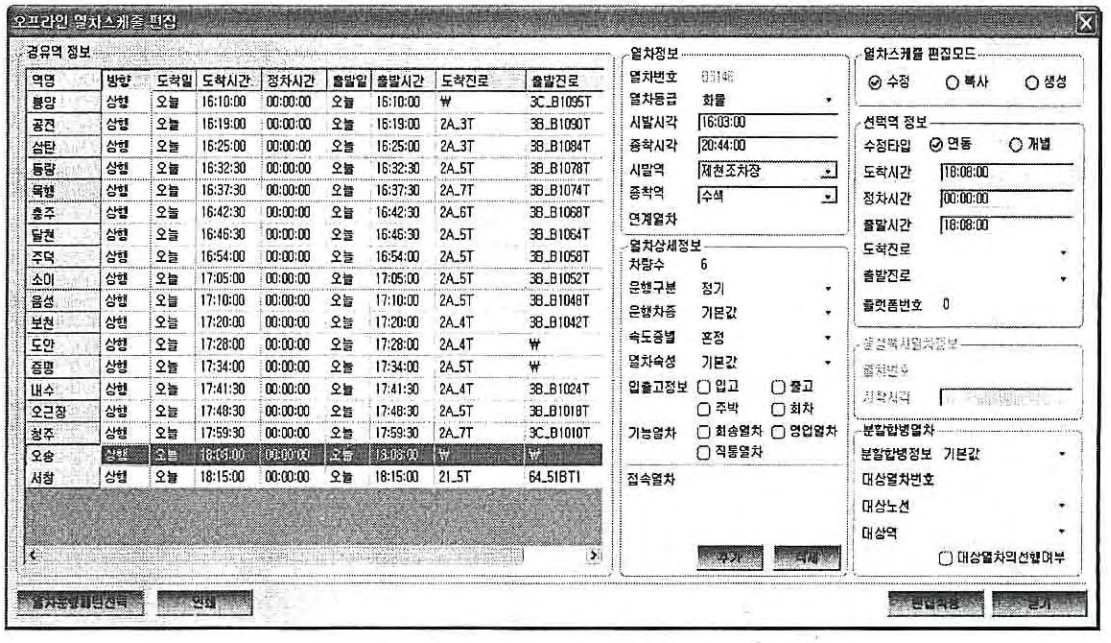


그림 3-64 오프라인 열차스케줄 편집

(마) 스케줄 관리 화면 - 오프라인 열차스케줄 ID간 열차복사

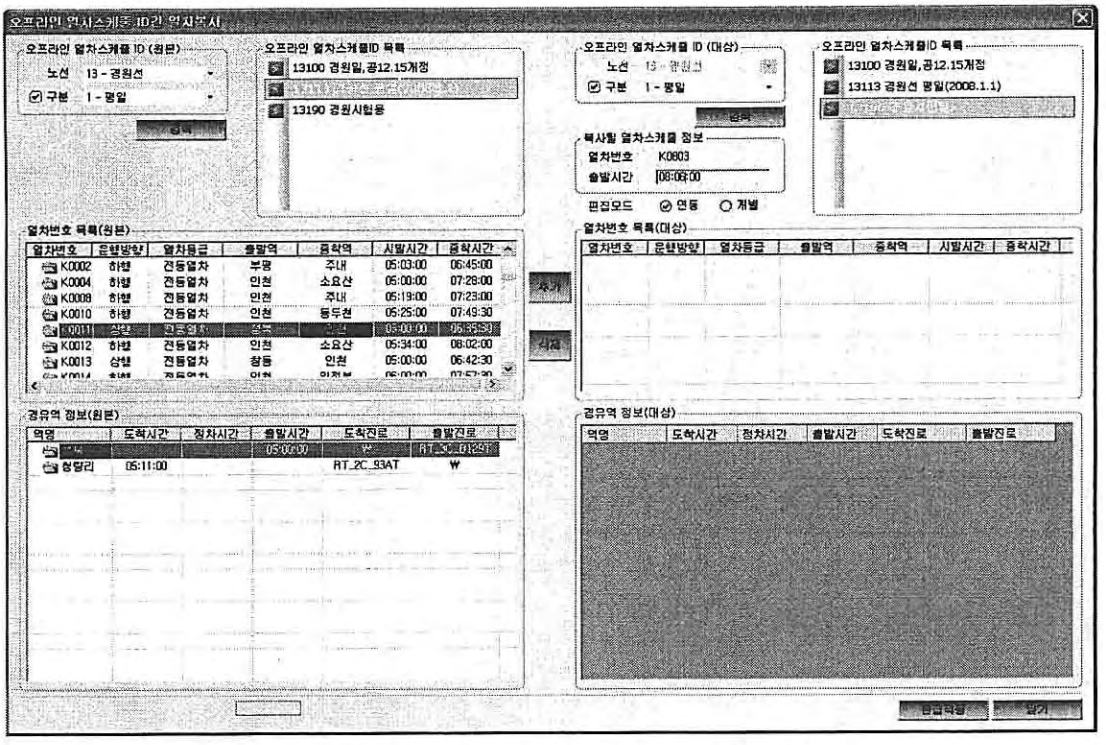


그림 3-65 오프라인 열차스케줄 ID간 열차복사

(바) 스케줄 관리 화면 - 온라인 열차스케줄 관리

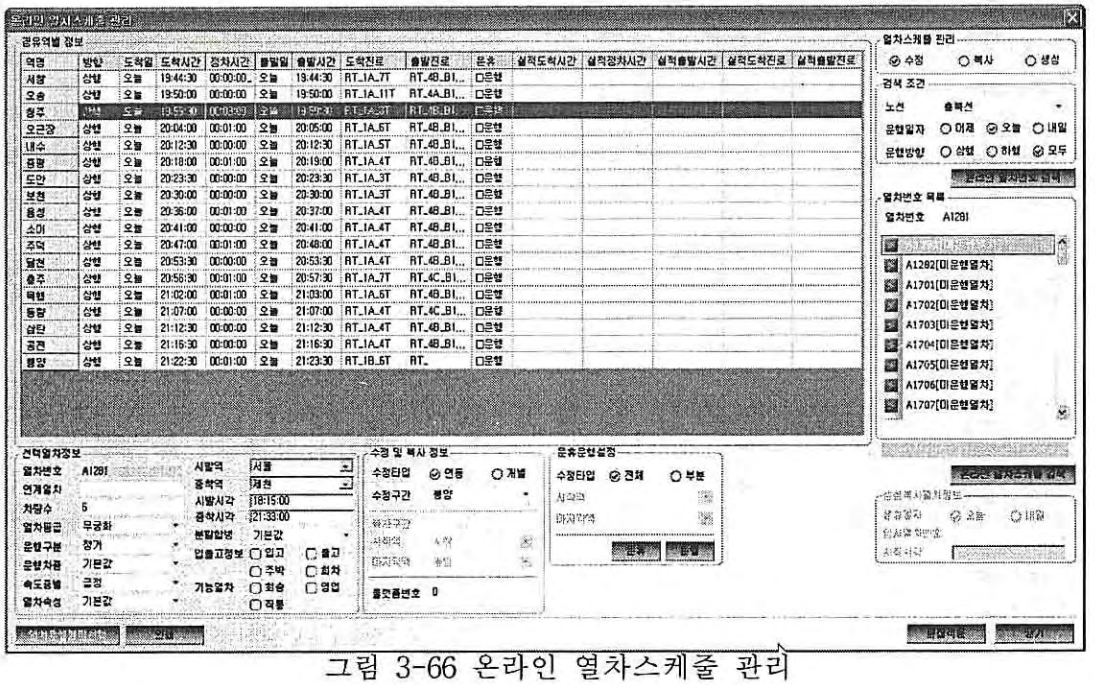


그림 3-66 온라인 열차스케줄 관리

2. 라인경합관리 서브시스템

가. 라인경합관리 서브시스템 상세설계 및 구축

라인경합관리 서브시스템(GCDRS; Global Conflict Detection & Resolution Subsystem)은 복수개의 역을 포함하는 서브네트워크의 열차운행관리 최적화를 위한 시스템으로 노선의 열차스케줄과 운행 중인 열차운행상태정보로부터 향후 일정한 시간 내에 발생할 경합을 예측하고 이를 해소한 후 열차지연을 최소화하는 본선 변경스케줄을 라인정보 서브시스템으로 전송하는 역할을 한다.

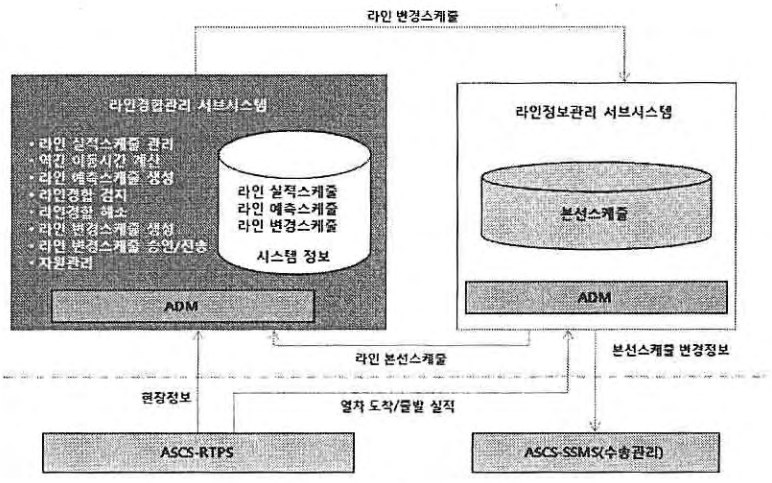


그림 3-67 라인경합관리 서브시스템

(1) 라인 열차스케줄 정의

(가) 라인 열차스케줄 전이도

- 라인정보 서브시스템에 있는 전체 본선스케줄에서 현재 시각부터 일정시간 이후까지(보통 1시간)의 본선스케줄을 가져온다. 이 본선스케줄이 GCDRS에서는 라인본선스케줄이 된다.
- 라인본선스케줄에 있는 도착/출발 실적 정보를 이용하여 실적 역간 이동시간을 계산한다.
- "열차예측계수계산 로직"을 이용하여 열차예측계수를 산출하고, 이 열차예측계수를 이용하여 다음역/이후역의 라인예측스케줄을 생성한다.

- 현재시간에서 일정시간 이내의 라인예측스케줄에서 라인경합검지를 수행한다.
- 경합이 검지되면 경합해소 알고리즘에 따라 라인변경스케줄을 생성한다.
- 라인관리자로부터 승인이 있으면 라인정보 서브시스템으로 라인변경스케줄을 전송한다.

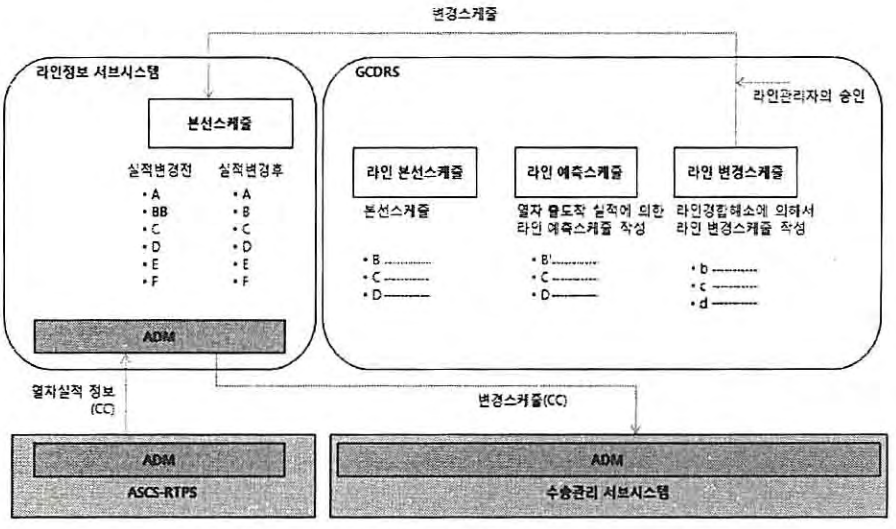


그림 3-68 라인경합관리 스케줄 전이도

(나) 라인본선스케줄

라인본선스케줄은 라인정보 서브시스템에 있는 전체 라인본선스케줄에서 현재시간 이후의 일정시간 내(보통 30분~1시간)에 있는 실적이 반영된 본선스케줄로, GCDRS는 파라메타로 설정한 주기마다 라인정보 서브시스템에 있는 본선스케줄을 가져온다.

(다) 라인예측스케줄

라인예측스케줄은 라인본선스케줄 중에서 열차 도착/출발 실적을 바탕으로 라인 역간 이동시간과 열차예측계수를 적용하여 예측한 스케줄로, 데이터베이스의 속성은 라인본선스케줄과 같다.

(라) 라인변경스케줄

라인변경스케줄은 라인예측스케줄의 변경으로 다음 역/이후 역 대해서 영향을 받는 모든 본선열차의 스케줄을 변경한 것으로, 데이터베이스의 속성은 본선스케줄과

같다.

(2) 세부 기능 정의

(가) 메시지 송수신

① 외부장치로의 데이터 송신

- 라인변경스케줄을 라인정보 서브시스템의 데이터베이스에 반영한다.

② 외부장치로부터의 데이터 수신

- 라인경합에 영향을 끼치는 연동정보를 CC형태로 수신한다.(열차고장, 궤도고장, 신호기 고장, 선로전환기 고장 등)
- 역의 TMS내 HR로부터 열차 순서 변경정보를 CC형태로 수신한다.
- 라인정보 서브시스템에 있는 데이터베이스의 본선스케줄을 읽어온다.

(나) 경보/이벤트 관리

o 라인경합관리 운영 중에 발생하는 각종 경보 및 이벤트를 기록하고 관리하는 기능이다. 세부 처리 내용은 다음과 같다.

- 경보/이벤트 파라메타 파일로부터 해당 정보를 읽어온다.
- 내부 메일박스로부터 데이터를 수신한다.
- 수신한 데이터를 이용하여 경보, 이벤트의 문구를 생성한다.
- 수신한 데이터와 문구를 데이터베이스에 기록한다.
- 생성한 경보/이벤트 문구를 화면에 표시하도록 관련(사용자 인터페이스) 프로세스로 전송한다.

(다) 라인예측스케줄 관리

- 라인본선스케줄 데이터베이스에서 열차 실적정보를 가져온다.
- 열차실적 도착 및 출발정보가 있으면, 실적 역간이동시간을 계산한다.
- 이전 역들의 열차실적을 이용해서 다음 역/이후 역에 대한 열차예측계수를 산출한다. ("열차예측계수계산 로직" 이용)
- 계산된 열차예측계수를 이용하여 다음 역/이후 역 역간이동시간을 계산한다. (라인본선스케줄에 있는 역간이동시간 * 열차예측계수)

- 계산한 실적 역간이동시간을 이용해서 해당 역 출발시간 및 다음 역 도착시간을 계산하여 예측시간으로 기록한다. 반복적으로 이후 역에 대해서도 출발/도착시간을 수정한다.
- 역 ASCS으로부터 열차출발 순서 변경요청을 수신시, 변경요청을 받은 역의 출발시간과 다음 역 도착시간을 계산하여 예측시간을 수정하고, 반복적으로 이후 역에 대해서도 출발/도착시간을 수정한다.
- 예측스케줄이 변경되면, 경보를 발생하여 라인경보관리 프로세스로 해당 경보내용을 전송한다.

(라) 라인경합검지

○ 현재시간에서 30분~1시간 이내에 있는 라인예측스케줄에 대해서 열차의 경합상황이 있는지 검사한다. 라인경합이 발생한 첫 번째 경합에 대해서 경합정보를 생성하여 MMI에 표시한다.

- 라인예측스케줄에서 현재시간 30분~1시간 이내의 변경이 발생하면, "라인경합검지 라이브러리"를 이용하여 라인경합을 검지한다.
- 라인경합이 검지되면, 검지한 첫 번째 경합상황을 경합으로 간주한다.
- 라인경합검지 내용을 MMI에 표시하기 위해서 해당 정보를 전송한다.
- 라인경합이 검지되면, 라인경합해소를 수행하도록 한다.
- 라인경합검지에 대한 경보를 발생하고, 라인경보관리 프로세스로 해당 경보내용을 전송한다.

(마) 라인경합해소

- 라인경합검지 정보가 있으면, "라인경합해소 라이브러리"를 이용해서 경합해소를 실행한다.
- 라인경합해소 결과에 의해서 라인변경스케줄을 작성한다.
- 라인경합정보를 MMI에 표시하여, 라인관리자가 라인경합해소 및 변경스케줄을 승인하도록 요청한다.
- 라인경합해소에 대한 경보를 발생하고 라인경보관리 프로세스로 해당 경보내용을 전송한다.

(바) 라인변경스케줄 기록

○ 라인경합해소 결과로 만들어진 라인변경스케줄을 전송하기 전에, 라인관리자의 승인을 받는다.

- MMI에서 라인관리자가 라인변경스케줄을 승인하면, 라인정보 서브시스템으로 변경스케줄을 전송한다.
- 라인변경스케줄로 데이터베이스 수정 후에, 라인변경스케줄의 내용을 라인예측스케줄에 반영한다.
- 데이터베이스의 수정에 대한 경보를 발생하고 라인경보관리 프로세스로 해당 경보내용을 전송한다.

(사) 사용자 인터페이스(MMI)

○ MMI는 라인관리자에게 다음과 같은 정보를 제공한다.

- 라인경합검지 및 해소에 대한 정보를 표시한다.
- GCDRS를 운영 중에 발생한 각종 경보 및 이벤트를 표시한다.
- 발생한 경보 및 이벤트의 확인을 수행하며, 데이터베이스에 저장된 내용을 조회할 수 있는 수단을 제공한다.
- 열차 DIA그래프를 이용하여 라인경합 상황을 쉽게 확인할 수 있도록 한다.
- 라인경합검지 정보를 수신하면, 해당 열차의 실적스케줄과 예측스케줄을 화면에 표시하여 경합상황을 표시한다.
- 시스템 환경을 관리하는 화면을 표시하고, 환경 변수들을 데이터베이스에 반영한다.
- 열차예측 범위 설정(현재시간 이후 열차를 관리해야 할 시간)
- 경합해소 목적함수 설정 (추가정차 회수 최소화, 한도초과 횟수 최소화)
- 열차 종별 가중치
- 여유시격 정보(도착, 출발, 교행)
- 역간 정보(역간 거리, TPS별 이동시간 상/하한 값, 열차유형, 라인특성)

○ MMI가 수신하여 처리하는 데이터의 종류는 다음과 같다.

- GCDRS 운영 중에 각 프로세스가 전송한 경보 메시지.
- 라인경합해소 실행 결과

- 라인본선스케줄과 라인변경스케줄 표시요구 정보
- 역의 TMS가 전송한 열차순서 변경 정보(HR 경합해소 결과)

○ CC형식으로 MMI가 외부장치로 송신하는 데이터는 다음과 같다.

- 승인한 라인변경스케줄을 라인정보 서브시스템에 전송한다.

○ MMI가 GCDRS 데이터베이스 기록하는 내용은 다음과 같다.

- 라인본선스케줄에 30분~1시간 이내 열차 추가.
- 라인예측스케줄 데이터베이스에 도착/출발 예측시간 수정
- 시스템 데이터베이스에 환경설정 파라메타 기록(열차예측 범위 설정, 경합해소 목적함수,
- 경보 데이터베이스에 경보/이벤트 추가
- 경보/이벤트의 확인 여부

(아) 라인변경스케줄 전송

○ MMI에서 라인관리자가 라인변경스케줄을 승인하면, 라인변경스케줄을 라인정보 서브시스템에 전송한다.

- 라인관리자가 라인변경스케줄을 승인하도록, MMI화면에 라인변경스케줄 정보를 표시한다.
- MMI화면에 정보를 표시한 후에, 승인이 없을 경우를 대비하여 타이머를 구동한다.
- 타이머가 만료되면, 라인변경스케줄 전송을 취소한다.
- 라인관리자의 승인이 있으면, 라인변경스케줄을 라인정보 서브시스템에 전송하고, 라인변경스케줄을 라인예측스케줄 데이터베이스에 반영한다.
- 라인변경스케줄 Update나 취소에 대한 경보를 발행하여, 라인경보관리 프로세스로 해당 경보내용을 전송한다.

(3) 라인경합관리 서브시스템 인터페이스 사양

(가) 라인정보 서브시스템과의 인터페이스

○ GCDRS가 라인정보 서브시스템과 인터페이스 하는 내용은 다음과 같다.

- 라인정보 서브시스템상의 데이터베이스로부터 본선스케줄을 주기적으로 라인 본선스케줄로 만든다.
- 라인본선스케줄로부터 열차실적정보를 읽고, 실적정보의 상태에 따라 예측/변경스케줄을 관리한다.

(나) 역의 실시간 열차위치추적 서브시스템(ASCS-RTPS)과의 인터페이스

○ GCDRS가 ASCS-RTPS와 인터페이스 하는 내용은 다음과 같다.

- CC의 형태로 현장정보(고장정보)를 수신한다.

(다) 역의 열차모니터링 서브시스템(ASCS-TMS)과의 인터페이스

○ GCDRS가 ASCS-TMS와 인터페이스 하는 내용은 다음과 같다.

- 라인 영역내의 ASCS에서 홈 리스케줄링의 홈 리스케줄링의 결과로, 열차출발순서가 변경되었을 때 변경정보를 CC형태로 수신한다.

(4) 서브프로세스 정의

○ 4개의 프로세스를 이용하여 GCDRS의 기능을 수행한다.

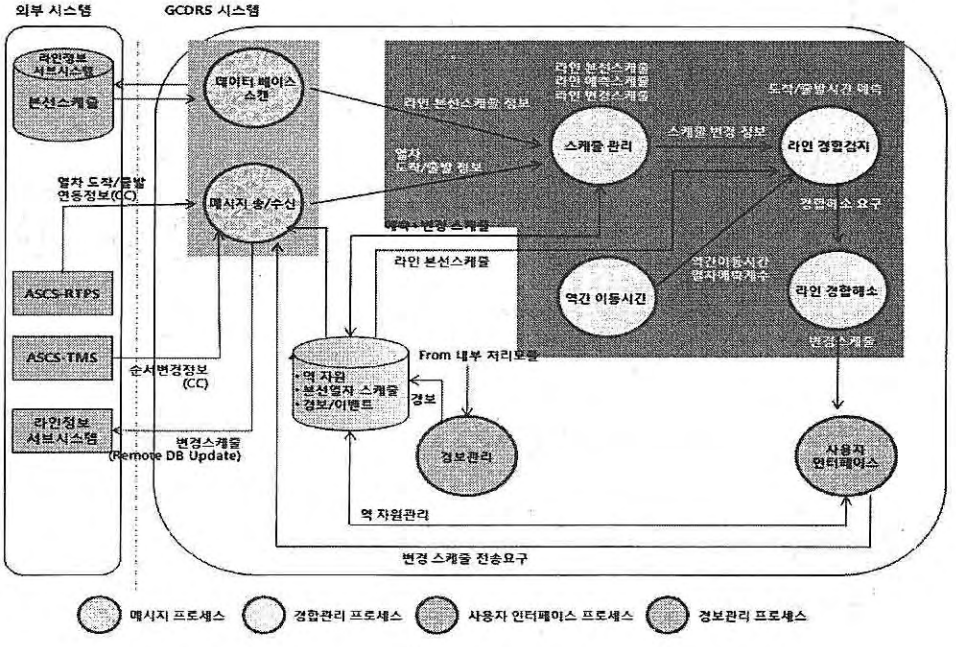


그림 3-69 라인경합관리 서브시스템

표 3-29 라인경합관리 서브시스템 기능

ID	프로세스	이름	기능
1	메시지 관리	TASK_MESSAGE	· 데이터베이스 스캔, 메시지 송/수신 처리
2	경합관리	TASK_RESCHEDULE	· 열차실적, 스케줄 관리, 라인경합검지, 라인경합해소
3	사용자 인터페이스	TASK_MMI	· 사용자 인터페이스
4	라인경보관리	TASK_ALARM	· 라인경보관리

(가) 메시지 관리 프로세스

① DFD

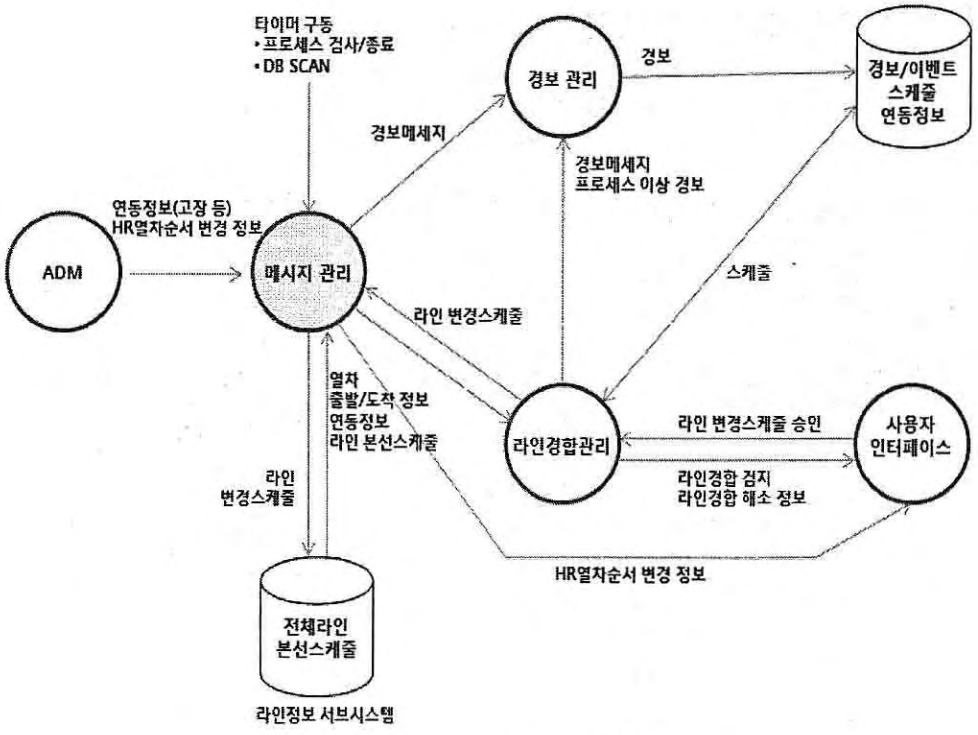


그림 3-70 GCDRS 메시지관리 DFD

② 송신데이터

표 3-30 GCDRS 메시지 관리 송신데이터

메시지 종류	To	설명
경보/이벤트 메시지	라인경보관리	· 수신한 메시지에 대한 이벤트 사항을 기록하기 위해서 메시지를 전송한다.
HR 열차순서 변경 정보	사용자 인터페이스	· HR에서 경합해소 결과, 열차의 순서가 변경되었다는 정보 수신시, MMI에 해당 정보를 표시한다.
연동정보(고장 정보)	라인경합관리	· 라인경합검지의 자료로 사용할 정보로, 라인경합관리에 의해서 해당 데이터베이스에 기록된다.
열차 실적 도착/출발 정보	라인경합관리	· 열차 실적 도착/출발 정보가 있으면, 변경 정보를 실적 및 예측스케줄에 반영하도록 요구한다.

③ 수신 데이터

표 3-31 GCDRS 메시지 관리 수신데이터

메시지 종류	From	설명
열차 실적 도착/출발 정보	자율분산관리자	· 열차 실적 도착/출발 정보
연동정보(고장 정보)	자율분산관리자	· 라인경합검지의 자료로 사용할 정보.
HR 열차순서 변경 정보	자율분산관리자	· HR에서 경합해소 결과, 열차의 순서가 변경되었다는 정보

(나) 경합관리 프로세스

① DFD

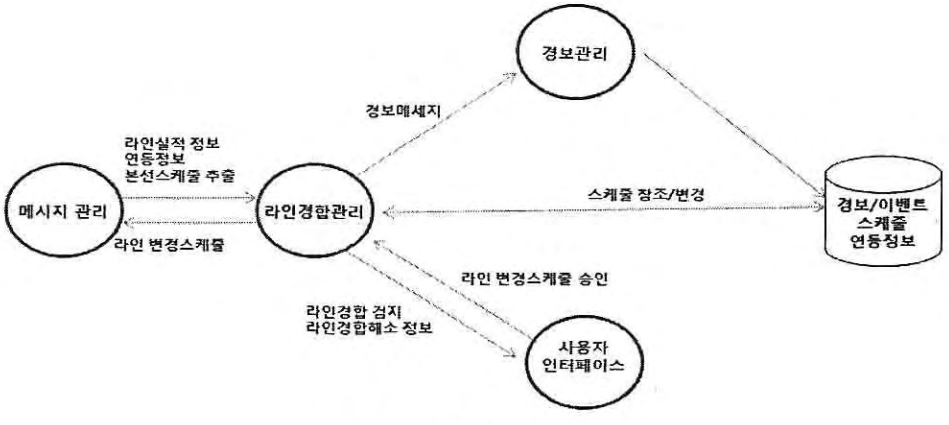


그림 3-71 라인경합관리 DFD

② 송신데이터

표 3-32 GCDRS 라인경합관리 송신데이터

메시지 종류	To	설명
라인경합검지 및 라인경합해소 정보	사용자 인터페이스	· 라인경합검지 결과나 경합 해소결과를 MMI에 표시하도록 요구한다.
경보메세지	경보 관리	· 경합관리에서 발생한 각종 경보/이벤트정보를 전송한다.
라인변경스케줄 승인요구	사용자 인터페이스	· 라인변경스케줄 승인요구

③ 수신 데이터

표 3-33 GCDRS 라인경합관리 수신데이터

메시지 종류	From	설명
라인 실적 정보	메시지 관리	· 라인본선스케줄에서 실적에 변경이 있는지 검사한다. · 라인 열차실적을 관리한다. · 라인예측스케줄을 수정한다. · 라인경합검지를 수행한다. · 역간이동시간을 계산한다. · 역간이동시간과 열차예측계수를 이용하여 다음역/이후역의 도착/출발 예측시간 계산한다. · 라인경합해소 및 라인변경스케줄 작성한다. · 라인변경스케줄 승인을 요청한다.
라인변경스케줄 승인	사용자 인터페이스	· 라인변경스케줄을 라인정보 서브시스템에 전송 한다.
라인예측스케줄 변경	사용자 인터페이스	· 사용자에 의한 라인예측스케줄 수동변경요구.
라인경합검지 요구	사용자 인터페이스	· 수동으로 라인예측스케줄 변경 시, 다시 라인경합을 검지한다.

(다) 사용자 인터페이스프로세스

① DFD

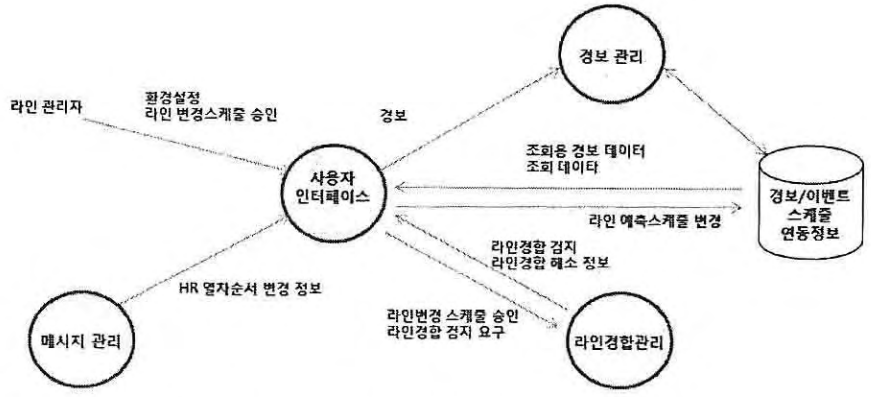


그림 3-72 GCDRS 사용자 인터페이스 DFD

② 송신 데이터

표 3-34 GCDRS 사용자 인터페이스 송신데이터

메시지 종류	To	설명
라인변경스케줄 승인	라인경합관리	· 라인관리자가 라인변경스케줄을 승인하면, 승인결과를 라인경합관리로 전송한다.
라인예측스케줄 변경정보	데이터베이스 라인경합관리	· GCDRS내의 라인예측스케줄을 수동으로 변경하고, 그 정보를 라인경합관리로 전송한다.
라인경합검지 요구	라인경합관리	· 수동으로 라인예측스케줄 변경 시, 다시 라인경합을 검지하도록 요구한다.
경보	라인경보관리	· 운영 중 발생한 경보를 저장 한다.

③ 수신 데이터

표 3-35 GCDRS 사용자 인터페이스 수신데이터

메시지 종류	From	설명
사용자 수동 조작	라인관리자	· 라인관리자로부터 다음과 같은 입력을 받는다. - 라인예측스케줄 수동 수정 - 환경설정 정보(역간정보, 열차예측계수, 여유시격 등)

		- 라인변경스케줄 승인
라인경합검지 정보	라인경합관리	· 라인경합검지 결과를 수신하여 해당 내용을 화면에 표시한다.
라인경합해소 정보	라인경합관리	· 라인경합해소 결과를 수신하여 해당 내용을 화면에 표시한다.
HR 열차순서 변경 정보	메시지 관리	· HR에서 경합해소 결과로 열차의 순서가 변경되었다는 정보수신시, 라인관리자에게 정보를 표시한다.
조회 데이터	데이터베이스	· 데이터베이스로부터 각종 조회 데이터를 읽어 화면에 표시한다. - 경보/이벤트 - 라인예측스케줄 - 라인변경스케줄 - 시스템 환경 정보 - 라인경합검지 및 해소 정보
라인변경스케줄 승인요구	라인경합관리	· 라인변경스케줄 승인요구
경보 문구	라인경보관리	· 화면에 표시할 경보 텍스트

(라) 라인경보관리 프로세스

① DFD

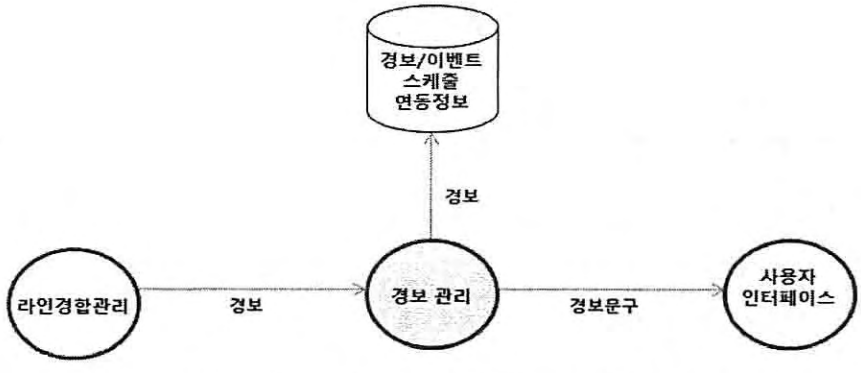


그림 3-73 GCDRS 라인경보관리 FLOW

② 송신 데이터

표 3-36 GCDRS 라인경보관리 송신데이터

메시지 종류	To	설명
경보문구	사용자 인터페이스	· 각 프로세스로부터 수신한 경보를 분석하여, MMI에 표시할 문구를 생성하여 MMI에 표시하도록 한다.
경보	데이터베이스	· 각 프로세스로부터 수신한 경보를 데이터베이스에 저장한다.

③ 수신 데이터

표 3-37 GCDRS 라인경보관리 수신데이터

메시지 종류	From	설명
경보	라인경합관리	각 프로세스로부터 수신한 경보를 분석하여 경보문구를 생성하여 MMI로 전송하고, 데이터베이스에 저장한다.
	사용자 인터페이스	
	메시지 관리	

(5) 경보/이벤트 정의

(가) 경보/이벤트 관리

사용자 인터페이스 프로세스는 라인경보관리 프로세스가 타 프로세스로부터 수신한 경보를 분석하여 만든 경보문구를 수신하여, MMI화면의 "이벤트 로그"에 표시한다.

① 경보문구 구성

발생한 경보는 발생시각, 경보종류, 등급 그리고 내용으로 구성된다.

경보문구 : 발생시각V경보종류V등급V내용 (V:space)

표 3-38 경보문구 구성

항목	내용	크기(Byte)
발생시각	"HH:MM:SS"의 24시 형식으로 표시한다.	8

등급	"B":Block, "M":Major, "m":minor 경보종류가 "EVENT"인 경우에는 Space로 한다.	1
역번호	역번호가 있는 경우에 역번호를 표시한다.	6
내용	텍스트	80

② 경보 표시 방법

- 경보는 검정색으로 표시한다.
- 가장 최근의 경보를 리스트 박스의 하단에 표시한다.
- 리스트박스에 경보가 차면, 경보내용을 스크롤하여 표시한다.
- 화면에 표시하는 경보의 개수는 기본 100개로 하며(파라메타로 변경), 가장 오래된 경보를 화면상에서만 자동으로 삭제한다.

(나) 경보/이벤트의 종류 및 등급

① 경보의 종류 및 등급

경보는 GCDRS운영 중에 발생하는 각종 장애나 오류를 나타나며, 주요 경보내용은 다음과 같다.

표 3-39 경보 종류 및 등급

분류	종류	등급
스케줄	- 라인본선스케줄 읽기 오류 - 라인예측스케줄 작성 오류 - 라인변경스케줄 작성 오류	Major
경합	- 라인경합검지 오류 - 라인경합해소 오류	Major
취급 조작	- 취급 오류	Minor
운영	- 프로세스 DOWN - 통신(NETWORK) 오류	Block
	- 송수신 메시지 오류	Major
고장	- 신호설비 고장 정보(신호기, 궤도, 선로전환기, 차량 등)	Major

② 이벤트의 종류

이벤트는 GCDRS운영 중에 발생하는 경보를 제외한 처리 상태나 사건 및 사용자 취급을 나타내며, 주요 이벤트의 내용은 다음과 같다.

표 3-40 이벤트의 종류

분류	종류
스케줄	- CC(Content Code) 데이터 수신 - 라인본선스케줄 SCAN - 라인예측스케줄 작성 - 라인변경스케줄 작성
경합	- 라인경합검지 - 라인경합해소
취급조작	- 데이터베이스 환경 설정 - 라인변경스케줄 승인 - 기타 중요 MMI 조작 사항 - 라인예측스케줄 수동 수정
운영	- 프로세스 구동

나. 라인경합 검지 및 해소 알고리즘 개발

(1) 수리모형

라인경합서브시스템에서는 라인영역내의 본선열차에 대하여 지연 및 외란 발생시 열차종별 우선순위에 따른 열차의 우선순위, 실제 출 도착 실적을 반영하여 추월 및 대피 순서와 위치를 최적 결정하는데 적용되는 열차경합해소문제를 이진(Binary)변수와 연속(continuous)변수가 혼합된 혼합정수계획법(Mixed Integer Programming)으로 모형화 하였다.

(가) 기호

- 본 연구에서 사용하는 수리모형에 대한 집합, 입력파마미터, 결정변수는 다음과 같다.

① 집합

- I : 열차집합

- P : 구간(segment)의 집합, n_p: 구간 $p \in P$ 에서의 선로의 개수
- Q : 역(station)의 집합
- I_p : 구간 p를 지나는 열차의 집합

② 입력 파라미터

- c_i : 열차 i의 출발 지연 비용
- w_i : 열차 i의 조착 또는 도착 지연 비용
- d_p^i : 열차 i의 구간 p에서 계획된 출발 시각
- a_p^i : 열차 i의 구간 p에서 계획된 도착 시각
- h_q : q역으로 진입 또는 q역에서 나갈 때 같은 선로를 사용하는 열차간의 여유시격
- h_m : 모든 역에서 착·발시 최소여유시격
- s_q^i : 역 q에서 열차 i의 최소 정차 시간
- $\underline{t}_p^i(oo)$:열차 i가 구간 p에서 oo형태로 이동할 때 걸리는 시간의 최솟값 $(o = s \text{ or } p)$
- $\overline{t}_p^i(oo)$: 열차 i가 구간 p에서 oo형태로 이동할 때 걸리는 시간의 최댓값
- b_p^i : 열차 i의 구간 p에서 출발할 수 있는 가장 빠른 시각

③ 결정변수

- D_p^i : 열차 i의 구간 p에서의 출발 시각
- A_p^i : 열차 i의 구간 p에서의 도착 시각
- E_p^i: 열차 i의 구간 p에서의 조착 또는 도착지연 시간
- $Z_{p,k}^i$: 열차 i가 구간 p의 k번째 선로를 지날 경우는 1, 아니면 0
- X_p^{ij} : 구간 p에서 열차 i가 j보다 먼저 출발하면 1, 아니면 0
- Y_p^{ij} : 구간 p에서 열차 i가 j보다 먼저 도착하면 1, 아니면 0
- OO_p^i : 구간 p에서 열차 i가 OO형태로 이동하면 1, 아니면 0$(O = S \text{ or } P)$

(나) 목적함수

- 목적함수는 출발 지연시간과 도착지연 혹은 조착 시간에 열차의 중요도에 따

라 가중치를 곱한 후, 이들의 합을 최소화하는 것이다.

$$\min \sum_{p\in P} \sum_{i\in I_p} c_i (D_p^i - d_p^i) + w_i E_p^i$$

(다) 제약식

① 도착시각의 지연 또는 조착시간

$$(A_p^i - a_p^i) \le E_p^i, \quad \forall p \in P, \ \forall i \in I_p$$
$$-(A_p^i - a_p^i) \le E_p^i, \quad \forall p \in P, \ \forall i \in I_p$$

② 열차 선로 선택(복선일 경우)

$$\sum_{k=1}^{n_p} Z_{p,k}^i = 1, \quad \forall p \in P, \ \forall i \in I_p$$

③ 동일선로를 운행하는 열차간의 시격 유지

$$M(1-Z_{p,k}^i) + M(1-Z_{p,k}^j) + \begin{cases} M(1-X_p^{ij}) + D_p^i - D_p^j \ge h_r \\ M(1-X_p^{ij}) + A_p^i - A_p^j \ge h_r \\ MX_p^{ij} + D_p^i - D_p^j \ge h_r \\ MX_p^{ij} + D_p^i - D_p^j \ge h_r \end{cases}$$
$$\forall p = (q,r) \in P, \ k = 1,\ldots,n_p, \quad \forall i,j \in I_p$$

④ 다른 선로를 운행하는 열차간의 시격 유지(동시착발제한)

$$\begin{aligned} &M(1-X_p^{ij}) + D_p^i - D_p^j \ge h_r, && \forall p = (q,r) \in P, \ \forall i,j \in I_p \\ &MX_p^{ij} + D_p^i - D_p^j \ge h_r, && \forall p = (q,r) \in P, \ \forall i,j \in I_p \\ &M(1-Y_p^{ij}) + A_p^i - A_p^j \ge h_r, && \forall p = (q,r) \in P, \ \forall i,j \in I_p \\ &MY_p^{ij} + D_p^i - D_p^j \ge h_r, && \forall p = (q,r) \in P, \ \forall i,j \in I_p \end{aligned}$$

⑤ 동일선로를 운행하는 열차의 출·도착순서 일치성

$$\begin{aligned} &2 - Z_{p,k}^i - Z_{p,k}^i + (X_p^{ij} - Y_p^{ij}) \ge h_m, \\ &2 - Z_{p,k}^i - Z_{p,k}^i - (X_p^{ij} - Y_p^{ij}) \ge h_m, \\ &\qquad \forall p = (q,r) \in P, \ \forall i,j \in I_p, \ k = 1,\ldots,n_p \end{aligned}$$

⑥ 각 역에서 열차 출발 시각 제약

$$D_p^i \geq b_p^i, \quad \forall p \in P, \ \forall i \in I_p$$

⑦ 구간 p에서 열차 i의 이동시간 제약

$$\underline{t_p^i}(ss)\,SS_i^p + \underline{t_p^i}(sp)\,SP_i^p + \underline{t_p^i}(ps)\,PS_i^p + \underline{t_p^i}(pp)\,PP_i^p \leq A_p^i - D_p^i,$$
$$\overline{t_p^i}(ss)\,SS_i^p + \overline{t_p^i}(sp)\,SP_i^p + \overline{t_p^i}(ps)\,PS_i^p + \overline{t_p^i}(pp)\,PP_i^p \geq A_p^i - D_p^i,$$
$$\forall p = (q,r) \in P, \quad \forall i \in I_p$$

⑧ 구간별 열차의 이동형태 선택

$$SS_p^i + SP_p^i + PS_p^i + PP_p^i = 1, \quad \forall p \in P, \ \forall i \in I_p$$

⑨ 역내 정차시간 준수

$$D_p^i - A_p^i \geq s_q^i,$$
$$(SS_p^i + SP_p^i) - 1 + \epsilon \leq D_p^i - A_p^i \leq M(SS_p^i + SP_p^i),$$
$$(SS_{p'}^i + PS_{p'}^i) - 1 + \epsilon \leq D_{p'}^i - A_{p'}^i \leq M(SS_{p'}^i + PS_{p'}^i),$$
$$\forall q = (p', p) \in Q, \ \forall i \in I_p, I_{p'}$$

제약 식을 설명하면 ①은 도착지연 또는 조착 시간을 의미하며 ②는 모든 열차가 각 구간에서 1개의 선로를 선택해야 하는 것을 뜻한다. ③은 동일한 선로를 운행하는 열차사이에 여유시격을 유지해야 함을, ④는 다른 선로(동일선로 포함)를 운행하는 열차사이에도 최소 여유시격이 지켜져야 하는 것을 뜻한다. ⑤는 동일한 선로를 운행하는 열차들의 경우, 출발과 도착의 순서가 동일해야 함을 의미한다. ⑥은 사고나 고장 등으로 정해진 출발시각을 지키지 못할 경우 출발이 가능한 가장 빠른 시각을 의미하는 것으로 경합을 표현하는 제약식이다. ⑦은 각 열차가 해당구간에서 정차패턴에 따라 정해진 속도를 준수해야 함을 뜻하며, ⑧은 모든 열차가 각 구간에서 하나의 정차패턴을 가져야 함을 의미한다. 마지막으로 ⑨는 모든 열차가 각 역에서 정해진 정차시간을 준수해야 함을 의미하는 제약 식으로, 정차패턴이 정차(stop)일 경우만 이를 지켜야 함을 의미한다.

(2) 수리모형의 특징 및 구조

(가) 수리모형의 특징

본 연구에서 다루고 있는 열차경합해소문제는 기존의 연구에서 다루었던 문제와 다음과 같은 다른 특징들을 가지고 있다. 우선 다른 선로를 운행하는 열차들 간에도 동시착발 제한으로 인해 출·도착 시에 일정한 여유시격을 유지해야 한다. 이때 여유시격이 동일선로를 운행하는 열차간의 여유시격과 다르기 때문에 2가지 여유시격 조건이 필요하다(제약식 ④, ⑤). 또한 각 구간에서 열차의 정차패턴(stop 또는 pass)에 따라 이동시간이 다르기 때문에 이를 반영하기 위해 Stop-Stop(SS), Stop-Pass(SP), Pass-Stop(PS), Pass-Pass(PP)의 4가지 경우를 고려한 이진변수를 추가하였다.

(나) 수리모형의 구조

열차경합해소문제의 구조를 자세히 보면 정차시간준수에 대한 제약식을 제외한다면, 문제가 정확히 각 구간별로 분리됨을 알 수 있다. 본 연구에서는 이렇게 각 구간별로 분리된 문제를 부문제(subproblem)이라고 부르고, 각 구간 $p=(q,r)\in P$에 대한 부문제의 해집합을 $X^p=\left\{x^p: H^p x^p \le d_p\right\}$ 로 정의한다. 또한 각 부문제를 연결하는 정차시간 준수 제약식을 $\sum_p J^p x^p \ge s$ 라고 한다면 열차경합해소문제는 다음과 같은 형태로 표현된다.

$$(IP)\quad z=\min\left\{\sum_p J^p x^p \ge s,\ x^p\in X^p,\ \forall p\in P\right\}$$

각각의 집합 X^p는 연속변수인 A와 D, E가 포함되어 있어서, 무한한 point를 가지고 있다. 그러나 출발시간과 도착시간의 경우, 각 열차간의 선후관계(X, Y)와 선로선택(Z)가 주어지면 쉽게 계산된다. 따라서 선후관계와 선로선택만을 고려한다면 X^p는 유한한 points를 가지고 있다고 가정할 수 있으며 $\left\{x^{p,t}\right\}_{t=1}^{T_p}$ 으로 쓸 수 있다. 이를 이용하면 X^p는 다음과 같이 표현된다.

$$X^p = \left\{ x^p : x^p = \sum_{t=1}^{T_p} \lambda_{p,t} x^{p,t},\ \sum_{t=1}^{T_p} \lambda_{p,t} = 1,\ \lambda_{p,t} \in 0,1 \right\}$$
$$\text{for } t = 1, \ldots, T_p$$

(*IP*)의 x^p를 위의 형태로 대체하면 다음의 IP Master Problem을 유도해 낼 수 있다.

$$(IPM) \qquad \begin{aligned} z &= \min \sum_p \sum_{t=1}^{T_p} (c_p x^{p,t}) \lambda_{p,t} \\ \sum_p \sum_{t=1}^{T_p} (J_p x^{p,t}) \lambda_{p,t} &\geq s \\ \sum_{t=1}^{T_p} \lambda_{p,t} &= 1 \\ \lambda_{p,t} &\in \{0,1\} \end{aligned}$$

위의 IPM에서 $\lambda_{p,t}$는 이진변수이다. 이를 LP로 완화한 문제를 LPM(Linear Programming Master Problem)이라고 한다.

$$(LPM) \qquad \begin{aligned} z^{LPM} &= \min \sum_p \sum_{t=1}^{T_p} (c_p x^{p,t}) \lambda_{p,t} \\ \sum_p \sum_{t=1}^{T_p} (J_p x^{p,t}) \lambda_{p,t} &\geq s \\ \sum_{t=1}^{T_p} \lambda_{p,t} &= 1 \\ \lambda_{p,t} &\geq 0 \end{aligned}$$

- 위의 LPM의 어려운 점은 변수인 $\lambda_{p,t}$의 개수가 기하급수적(exponential)으로 많다는데 있다. 열생성(Column generation) 알고리듬의 초기화 단계는 무수히 많은 변수들(λ) 중에서 일부를 선택하는 것이다. LPM에서 column들의 subset을 정의하고, 이 subset들로 이루어진 문제를 RLPM(Restricted Linear Programming Master Problem)이라 하자. 열생성 알고리듬은 다음의 과정을 수행한다.
- Initialization :

- RLPM을 정의하고 RLPM의 최적해 $\tilde{\lambda}^*$을 구한다.

○ Primal Feasibility:

- RLPM의 모든 가능해는 LPM의 가능해가 된다. 따라서 $\overline{z^{LPM}} \leq z^{LPM}$ 를 만족한다.

○ Optimality Check for LPM :

- 연결제약식(정차시간 제약식)에 해당하는 쌍대 변수를 π, 나머지 제약 식에서의 쌍대 변수를 μ라고 하자. 이 경우에 현재의 해가 LPM에서 최적이기 위해서는 모든 p에 대하여 reduced cost가 비음인 조건을 만족해야 한다. $(c^p x - \pi J^p x - \mu_p \geq 0)$. 비음 조건을 확인하기 위하여 다음과 같은 ζ_p를 정의한다.

$$\zeta_p = \min\left\{(c^p - \pi J^p)x - \mu_p : x \in X^p\right\}$$

○ Stopping criterion:

- 만약 모든 p에 대해 $\zeta_p \geq 0$이라면 해 (π, μ)는 LPM의 쌍대 가능해이고, 따라서 $\tilde{\lambda}^*$는 LPM의 최적해라고 할 수 있다.

○ Generating a New Column:

- 만약 어떤 구간 p에 대해 $\zeta_p < 0$이라면 이에 해당하는 행을 RLPM에 추가하여 위의 과정을 반복한다.
- 위의 열생성 알고리듬에서는 매번 LPM의 최적성을 검사하기 위해 부문제(subproblem)의 해를 구해야 한다. 즉, 정차시간 준수 제약식을 제외한 임의의 구간 1개에서의 부문제를 매우 빨리 풀 수 있어야 한다. 다음 장에서 LPM의 최적성을 검사하기 위해 부문제를 정확하게 규명하고, 휴리스틱 알고리듬을 사용한 부문제의 해법을 제시하겠다. 그리고 이를 이용하여 전체 문제의 해를 구하는 알고리듬을 제시한다.

(3) 알고리듬 개발

(가) 부문제(Subproblem) 정의

- 앞서 소개했던 부문제에 대하여 자세히 다루고자 한다. column generation 알

고리듬 과정에서 부문제는 다음과 같은 형태로 표현된다.

$$\zeta_p = \min\{(c^p - \pi J^p)x - \mu_p : x \in X^p\}$$

- $\overline{c_p} = c_p - \pi J^p$로 정의하면 부문제는 다음과 같은 최적화 문제가 된다.

$$\min \ \overline{c_D^p} D^p + \overline{c_A^p} A^p + \overline{c_E^p} E^p + (\overline{c_{ss}^p} SS^p + \overline{c_{sp}^p} SP^p + \overline{c_{ps}^p} PS^p + \overline{c_{pp}^p} PP^p) - \mu_p$$

s.t

$$(A_p^i - a_p^i) \le E_p^i, \quad \forall i \in I_p$$
$$-(A_p^i - a_p^i) \le E_p^i, \quad \forall i \in I_p$$

$$\sum_{k=1}^{n_p} Z_{p,k}^i = 1, \quad \forall p \in P, \ \forall i \in I_p$$

$$M(1 - Z_{p,k}^i) + M(1 - Z_{p,k}^j) + \begin{cases} M(1 - X_p^{ij}) + D_p^i - D_p^j \ge h_r \\ M(1 - X_p^{ij}) + A_p^i - A_p^j \ge h_r \\ MX_p^{ij} + D_p^i - D_p^j \ge h_r \\ MX_p^{ij} + D_p^i - D_p^j \ge h_r \end{cases}, \quad k = 1, \ldots, n_p, \quad \forall i, j \in I_p$$

$$M(1 - X_p^{ij}) + D_p^i - D_p^j \ge h_r, \quad \forall i, j \in I_p$$
$$MX_p^{ij} + D_p^i - D_p^j \ge h_r, \quad \forall i, j \in I_p$$
$$M(1 - Y_p^{ij}) + A_p^i - A_p^j \ge h_r, \quad \forall i, j \in I_p$$
$$MY_p^{ij} + D_p^i - D_p^j \ge h_r, \quad \forall i, j \in I_p$$

$$2 - Z_{p,k}^i - Z_{p,k}^i + (X_p^{ij} - Y_p^{ij}) \ge h_m, \ \forall i, j \in I_p, \ k = 1, \ldots, n_p$$
$$2 - Z_{p,k}^i - Z_{p,k}^i - (X_p^{ij} - Y_p^{ij}) \ge h_m, \ \forall i, j \in I_p, \ k = 1, \ldots, n_p$$

$$D_p^i \ge b_p^i, \quad \forall i \in I_p$$

$$\underline{t_p^i}(ss) SS_i^p + \underline{t_p^i}(sp) SP_i^p + \underline{t_p^i}(ps) PS_i^p + \underline{t_p^i}(pp) PP_i^p \le A_p^i - D_p^i, \ \forall i \in I_p$$
$$\overline{t_p^i}(ss) SS_i^p + \overline{t_p^i}(sp) SP_i^p + \overline{t_p^i}(ps) PS_i^p + \overline{t_p^i}(pp) PP_i^p \ge A_p^i - D_p^i, \ \forall i \in I_p$$

$$SS_p^i + SP_p^i + PS_p^i + PP_p^i = 1, \quad \forall i \in I_p$$

위의 부문제는 목적함수의 계수(coefficient)의 변화를 제외한다면, 한 구간에서의 열차경합해소문제를 푸는 것과 동일하다. 바뀐 목적함수를 살펴보자. 각 목적함수

들의 계수 값인 $\overline{c_p}$는 $\overline{c_p} = c_p - \pi J^p$로 이 값을 알기 위해서는 RLPM의 dual solution인 (π, μ) 값을 알아야 한다.

① $D_p^i - A_p^{i'} \geq s_q^i, \qquad \forall q=(p',p) \in Q,\ \forall i \in I_p, I_{p'}$

② $(SS_p^i + SP_p^i) - 1 + \epsilon \leq D_p^i - A_{p'}^i,\ \forall q=(p',p) \in Q,\ \forall i \in I_p, I_{p'}$

③ $D_p^i - A_p^{i'} \leq M(SS_p^i + SP_p^i), \qquad \forall q=(p',p) \in Q,\ \forall i \in I_p, I_{p'}$

④ $(SS_{p'}^i + PS_p^i) - 1 + \epsilon \leq D_p^i - A_{p'}^i,\ \forall q=(p',p) \in Q,\ \forall i \in I_p, I_{p'}$

⑤ $D_p^i - A_{p'}^{i'} \leq M(SS_{p'}^i + PS_p^i), \qquad \forall q=(p',p) \in Q,\ \forall i \in I_p, I_{p'}$

위의 정차시간준수 제약식들은 각 열차별로 역마다 존재한다. i 열차의 역 q에서의 제약식 (가)에 해당하는 dual solution을 $\pi_1^{i,q}$라 하자. 그러면 목적함수의 coefficient값은 다음과 같이 표현할 수 있다.

- Case 1. 원래 스케줄이 SS일 경우

$\overline{c_D^p} = c_i - \pi_1^{i,q},\ \overline{c_A^p} = \pi_1^{i,r},\ \overline{c_E^p} = w_i,\ \overline{c_{ss}^p} = \overline{c_{sp}^p} = \overline{c_{ps}^p} = \overline{c_{pp}^p} = 0$

- Case 2. 원래 스케줄이 SP일 경우

$\overline{c_D^p} = c_i - \pi_1^{i,q},\ \overline{c_A^p} = \pi_1^{i,r} + \pi_2^{i,r} - \pi_3^{i,r} + \pi_4^{i,r} - \pi_5^{i,r},\ \overline{c_E^p} = w_i$

$\overline{c_{ss}^p} = \pi_4^{i,r} - M \bullet \pi_5^{i,r},\ \overline{c_{ps}^p} = \pi_4^{i,r} - M \bullet \pi_5^{i,r},\ \overline{c_{sp}^p} = \overline{c_{pp}^p} = 0$

- Case 3. 원래 스케줄이 PS일 경우

$\overline{c_D^p} = c_i - \pi_1^{i,q} - \pi_2^{i,q} + \pi_3^{i,q} - \pi_4^{i,q} + \pi_5^{i,q},\ \overline{c_A^p} = \pi_1^{i,r},\ \overline{c_E^p} = w_i$

$\overline{c_{ss}^p} = \pi_2^{i,q} - M \bullet \pi_3^{i,q},\ \overline{c_{sp}^p} = \pi_2^{i,q} - M \bullet \pi_3^{i,q},\ \overline{c_{ps}^p} = \overline{c_{pp}^p} = 0$

- Case 4. 원래 스케줄이 PP일 경우

$\overline{c_D^p} = c_i - \pi_1^{i,q} - \pi_2^{i,q} + \pi_3^{i,q} - \pi_4^{i,q} + \pi_5^{i,q},\ \overline{c_A^p} = \pi_1^{i,r} + \pi_2^{i,r} - \pi_3^{i,r} + \pi_4^{i,r} - \pi_5^{i,r},\ \overline{c_E^p} = w_i$

$\overline{c_{ss}^p} = \pi_2^{i,q} - M \bullet \pi_3^{i,q} + \pi_4^{i,r} - M \bullet \pi_5^{i,r}, \qquad \overline{c_{sp}^p} = \pi_2^{i,q} - M \bullet \pi_3^{i,q}, \qquad \overline{c_{ps}^p} = \pi_4^{i,r} - M \bullet \pi_5^{i,r},$

$\overline{c_{pp}^p} = 0$

현재 목적함수에는 D^p, A^p, E^p 이외에 각 열차의 구간에서의 이동형태를 결정하는 변수 SS^p, SP^p, PS^p, PP^p도 포함되어 있다. 이 변수들은 모두 이진변수로 4개중

오직 한 개의 변수만이 1값을 가지며 나머지는 모두 0이 된다. 또한 변수들의 coefficient에는 모두 $-M$ 값이 포함되어 있어서 그에 해당하는 dual solution이 양수인 경우 매우 작은 값을 갖게 된다. 따라서 본 연구에서는 변수의 선택으로 D, A, E 값이 변해서 목적함수에 미치는 영향보다 실제 변수의 값이 목적함수에 미치는 영향이 크다고 가정하고, 각 변수들의 coefficient를 계산한 후 가장 작은 값을 갖는 변수를 1로 선택한다.

이러한 가정을 추가시키면 부문제는 다음과 같이 간단히 모형화가 가능하다.

$$\min \ \overline{c_D^p} D^p + \overline{c_A^p} A^p + \overline{c_E^p} E^p - \mu_p$$

s. t

$$(A_p^i - a_p^i) \le E_p^i, \quad \forall i \in I_p$$
$$-(A_p^i - a_p^i) \le E_p^i, \quad \forall i \in I_p$$

$$\sum_{k=1}^{n_p} Z_{p,k}^i = 1, \quad \forall p \in P, \ \forall i \in I_p$$

$$M(1 - Z_{p,k}^i) + M(1 - Z_{p,k}^j) + \begin{cases} M(1 - X_p^{ij}) + D_p^i - D_p^j \ge h_r \\ M(1 - X_p^{ij}) + A_p^i - A_p^j \ge h_r \\ MX_p^{ij} + D_p^i - D_p^j \ge h_r \\ MX_p^{ij} + D_p^i - D_p^j \ge h_r \end{cases}, \quad k = 1, \ldots, n_p, \quad \forall i, j \in I_p$$

$$M(1 - X_p^{ij}) + D_p^i - D_p^j \ge h_r, \quad \forall i, j \in I_p$$
$$MX_p^{ij} + D_p^i - D_p^j \ge h_r, \quad \forall i, j \in I_p$$
$$M(1 - Y_p^{ij}) + A_p^i - A_p^j \ge h_r, \quad \forall i, j \in I_p$$
$$MY_p^{ij} + D_p^i - D_p^j \ge h_r, \quad \forall i, j \in I_p$$

$$2 - Z_{p,k}^i - Z_{p,k}^i + (X_p^{ij} - Y_p^{ij}) \ge h_m, \ \forall i, j \in I_p, \ k = 1, \ldots, n_p$$
$$2 - Z_{p,k}^i - Z_{p,k}^i - (X_p^{ij} - Y_p^{ij}) \ge h_m, \ \forall i, j \in I_p, \ k = 1, \ldots, n_p$$

$$D_p^i \ge b_p^i, \quad \forall i \in I_p$$

$$\underline{t_p^i} \le A_p^i - D_p^i \le \overline{t_p^i}, \ \forall i \in I_p$$

이 부문제는 다음과 같은 스케줄링 문제와 유사한 문제로 설명될 수 있다. 선로를 독립적인 machine으로 보고 각 열차를 release time b_i, processing time $p_i (l_i \le p_i \le u_i)$, dependent setup time h_{ij}인 job으로 생각하여, total weighted completion time + total weighted start time + total weighted delay time을 최소

화시키는 스케줄링 문제가 된다. 그런데 스케줄링 문제와 다음의 요소가 다르다.

- job의 setup은 이전 job이 투입된 이후 바로 시작할 수 있다.
- setup이 완료되면 이전 job이 끝나기 전에 다음 job을 동시에 수행할 수 있다.
- 다른 machine을 사용하는 job간에도 release time이 존재한다.
- 음수의 weight가 존재할 수 있다.

열차경합해소문제는 구간이 1개인 경우에도 NP-hard로 어려운 문제이다. 따라서 위의 부문제 역시 원래 문제에 비해서 훨씬 쉬운 문제이긴 하지만, 열차의 개수가 많아지게 되면 계산시간이 급격히 증가하게 된다. 본 연구에서는 다음과 같은 부문제를 푸는 휴리스틱 알고리듬을 제시하였다.

(나) 부문제 해법

① 부문제 휴리스틱 알고리듬

각 열차를 가능한 모든 선로에 대해서 다음의 작업을 수행한 후에 penalty의 가중평균$(\overline{c_D^p} \cdot D + \overline{c_A^p} \cdot A)/(\overline{c_D^p} + \overline{c_A^p})$이 가장 작은 열차의 스케줄을 가장 먼저 배정한다.

- Step 0 :

열차의 출발 시각은 예정된 출발 시각으로 한다.

- Step 1 :

현재 배정된 열차와 출발역에서 여유시격을 만족시키지 못할 경우, 가장 빠르게 출발할 수 있는 시각으로 재조정하고, 도착 시각은 도착 지연이 없이 최대한 예정된 도착시각과 가깝게 정한다.

- Step 2 :

현재 배정된 열차와 도착역에서 여유시격을 만족시키지 못할 경우, 배정된 열차와의 출발 순서에 따라 도착 시각을 앞당기거나 뒤로 미룬 뒤에 가장 빠르게 도착할 수 있는 시각으로 정한다.

- Step 3 :

현재 도착 시각이 이동시간 제약을 만족하면 종료한다. 만약 이동시간이 제약의 하한을 만족하지 못할 경우 도착시각을 출발시간+하한으로 재조정 후 Step 2로 이동하고, 상한을 만족하지 못할 경우 상한을 만족하도록 출발시각을 재조정하고

Step 1로 돌아간다.

(다) Fix and Regenerate 알고리듬

① 해법의 요구사항 및 아이디어

이번 절에서는 앞 절의 부문제 휴리스틱 알고리듬을 바탕으로 실제 열차경합해소 문제에 대한 Fix and Regenerate 알고리듬을 소개한다. 먼저 해법의 요구사항을 살펴보면 다음과 같다. 문제의 제약식을 모두 만족하면서 지연 비용을 최소로 하는 스케줄을 제공해야 한다. 또한 경합이 발생했을 시 신속한 경합해소가 요구되므로, 최대한 빠른 시간 내에 문제의 해를 제공해야 한다.

해법의 아이디어를 간단히 설명하면 다음과 같다. 먼저 실제 문제를 풀고자 하는 수도권에서의 철도네트워크는 비교적 간단히 표현된다. 또한 앞서 설명했듯이 열차경합해소문제는 각 구간별로 분리할 수 있으며, 이렇게 분리된 부문제 휴리스틱 알고리듬을 이용하여 쉽게 풀 수 있다. 따라서 각 구간을 위상정렬(topological sort)하여, 각 부문제를 순차적으로 풀어나가는 방법으로 알고리듬을 개발하였다.

(4) Fix and Regenerate 알고리듬

(가) Fix and Regenerate 알고리듬

- Step 0 :

원래 문제에 대한 가능해를 하나 구하고, 열생성 알고리듬에서 정의하는 RLPM(Restricted LP Master problem)을 구성한다.

- Step 1 :

RLPM의 최적해를 구하고, 각 부문제를 휴리스틱으로 풀어서 reduced cost가 음수인 해를 열생성한다. 모든 부문제에 대해서 reduced cost가 비음일 때까지 이 과정을 반복한다.

- Step 2 :

정렬된 구간에서 순서가 가장 빠른 구간에 대한 해 중, RLPM의 해(λ) 중 가장 큰 λ를 1로 fix한다.

- Step 3 :

현재 fix된 해를 이용하여 남은 구간에 대해서 원래 문제의 가능해를 regenerate하고, 이를 이용하여 RLPM을 다시 구성한다. fix 되지 않은 구간이 있는 경우

Step 1로 돌아간다.

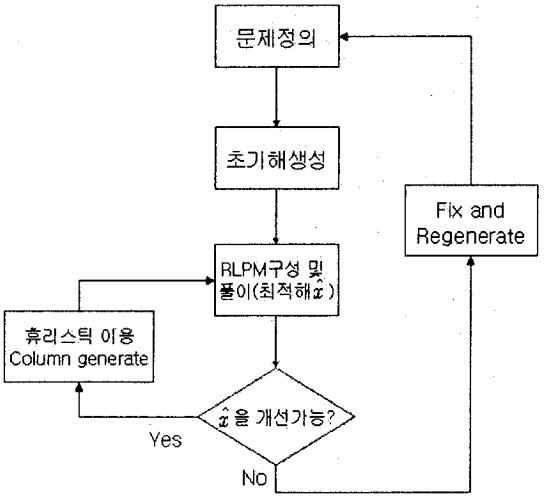


그림 3-74 알고리듬 전개 flow chart

3. 라인정보 서브시스템

가. 시스템 설계

대형·고밀도 역제어시스템 구축을 위해 라인정보 서브시스템(LIS; Line Information System)이 구현되며, 라인정보시스템은 CTC의 운행관리 기능 중 일부인 열차스케줄 작성, 열차 그래프 기능을 해당 라인에 대해서 수행하도록 어플리케이션이 작성된다. 라인정보 서브시스템은 대형·고밀도운행 철도역 자동운행제어시스템 중에서 센터에 위치하게 되는 라인센터 시스템의 서브시스템이며 열차의 운행관리를 담당한다.

라인정보 서브시스템은 역 ASCS와 DF를 통해 연결되어 오프라인 열차스케줄의 전송 및 온라인 열차스케줄의 변경 정보를 전송하는 역할을 하게 된다.

라인정보 서브시스템에서 관리되는 열차 운행계획은 오프라인 및 온라인에 대한 본선스케줄로 제한이 되며, 역 입환 스케줄에 대한 관리는 수행하지 않는다.

다음 그림 3-75는 라인정보 서브시스템 구축에 따른 ASCS 시스템의 구성도를 나타낸 것이다.

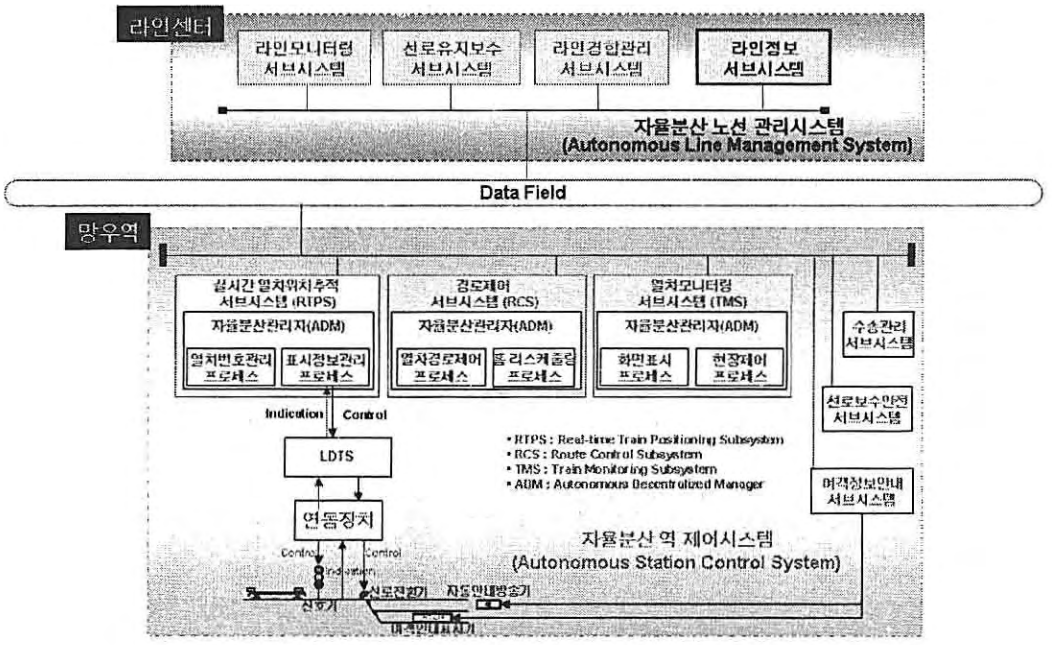


그림 3-75 라인정보 서브시스템 구성도

나. CTC 병용운행의 개념

자율분산 역제어시스템 및 라인정보시스템이 CTC와 병용운행 하는 경우는 자율분산 역제어시스템에 역에 투입되어 운영되는 각 단계별로 구별할 수 있다.

(1) 기존 역 변경에 따른 단독역 구축

첫 번째 단계는 열차집중제어장치의 제어 상태에 있는 역을 자율분산 역 제어시스템으로 변경하는 단계이다. 단일역에 설치된 자율분산 역 제어시스템의 역할은 다음과 같다.

- 제어 범위 : 해당역 제어구간
- 시스템 운영모드 : 역 단독 모드
- 스케줄 수신 : 역 단독 모드는 라인센터와 연결되지 않은 상태로 CTC 스케줄을 파일로 전송받아 역 운영자가 직접 스케줄을 TMS를 통해 입력하고 RCS가 해당역 제어구간의 열차가 진입하면 홈리스케줄링 프로세스에서 최적화를 실행한 후 자동 경로제어 명령을 RTPS로 전송하여 진로를 제어함
- 열차번호검지 : 단일역 설치 상황에서의 열차번호 검지는 역 제어구간에 진입한 열차에 대해서 RTPS에서 검지한 열차와 보유하고 있는 스케줄의 비교

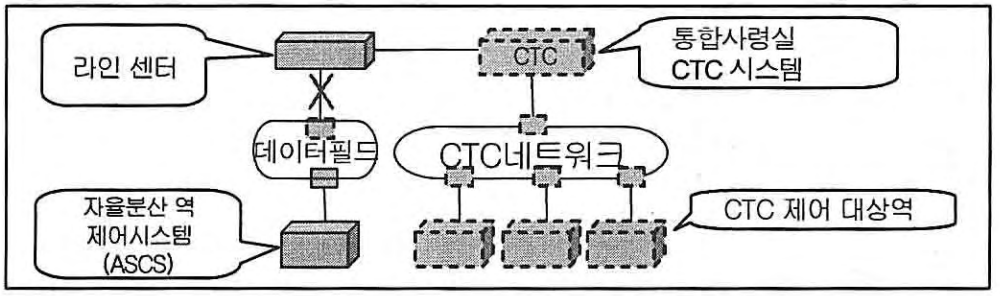


그림 3-76 역 단독 모드 구축 개념도

(2) 기존 역 변경에 따른 단일역 구축

단일역 모드는 첫 번째 단계인 역단독 모드에서 센터의 라인정보 시스템과 연결되어 있지 않은 구조에서 ADS 네트워크를 통해서 라인정보와 연결된 상태를 의미한다. 단일역 모드에서 자율분산 역 제어시스템의 역할은 다음과 같다.

- 제어 범위 : 해당역 제어구간
- 시스템 운영모드 : 단일역 운영모드
- 스케줄 수신 : 라인정보 시스템에서 일일 스케줄과 변경 스케줄을 수신 받아서 RCS가 해당역 제어구간의 열차가 진입하면 홈리스케줄링 프로세스에서 최적화를 실행한 후 자동 경로제어 명령을 RTPS로 전송하고 RTPS는 경로제어 명령을 연동장치 등으로 전달한다.
- 열차번호검지 : 단일역 설치 상황에서의 열차번호 검지는 역 제어구간에 진입한 열차에 대해서 RTPS에서 검지한 열차와 보유하고 있는 스케줄의 비교

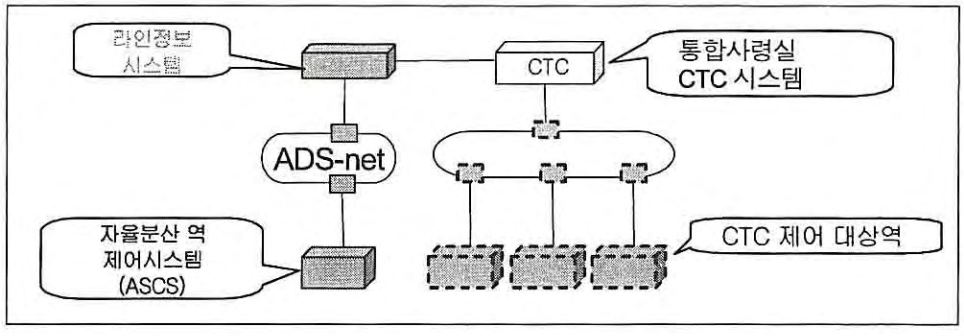


그림 3-77 단일역 구축 개념도

(3) 다중역 모드

단일역 자율분산 역 제어시스템 구축 후, 두 번째 확장단계는 인접역으로의 확장단계이다. 인접역 설치 확장에 따른 자율분산 역 제어시스템의 역할과 기능은 다음과 같다.

- 제어 범위 : 각 역별 제어구간
- 시스템 운영모드 : 다중역 운영모드
- 스케줄 수신 : 단일역 운영모드와 마찬가지로 라인정보 시스템에서 열차스케줄을 수신하며, RCS에서 자동 경로제어를 수행한다.
- 열차번호검지 : 단일역 모드에서의 열차번호 검지는 역의 제어구간 내에 진입한 열차에 대해 검지를 통한 스케줄 비교를 통해 운영하였으나 인접역 연동모드에서는 전역 자율분산 역 제어시스템의 RTPS에서 송신한 열차추적정보를 해당역의 RTPS에서 수신하여 운영한다.

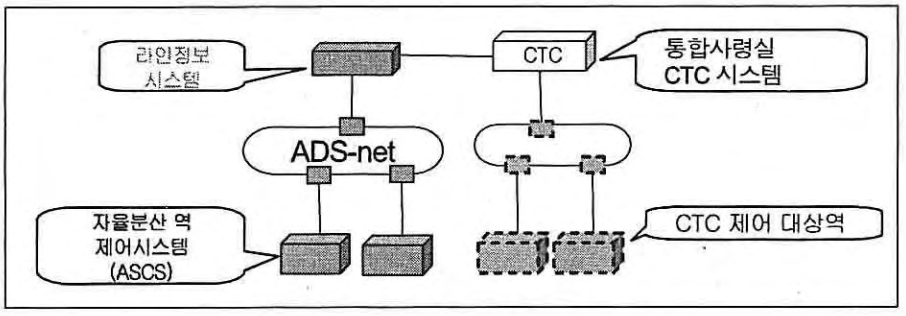


그림 3-78 인접역 확장 개념도

(4) 자율분산 노선관리 시스템 구축(Full ADTMS Mode)

자율분산 노선관리 시스템 구축이 노선단위의 열차운행관리 시스템의 최종단계이다. 이러한 노선단위 시스템 구축은 노선별 분리에 의한 많은 데이터 전송량을 줄일 수 있을 뿐만 아니라 데이터필드에서 노선과 관련이 없는 데이터를 효율적으로 줄일 수 있으며 다른 노선 네트워크에서 발생되는 고장의 영향에서도 벗어날 수 있기 때문이다.

자율분산 노선관리 시스템의 구축은 이전단계에서 CTC의 스케줄을 라인정보에서 수신하여 자율분산 역 제어시스템에 알맞게 수정해서 제공하였으나 자율분산

노선관리 시스템의 구축과 함께 라인정보시스템은 노선에 포함된 모든 역들의 열차스케줄을 생성하며 노선단위의 열차 운행 모니터링과 노선에서 발생되는 열차 경합문제를 해결 할 수 있다.

- 제어 범위 : 각 역별 제어구간
- 시스템 운영모드 : Full ADTMS Mode
- 스케줄 수신 : 자율분산 노선관리 시스템에서 라인정보 서브시스템으로부터 각 역의 열차 스케줄을 수신 받아 운영한다.
- 열차번호검지 : 센터 연동모드에서는 센터에서 수신한 열차 스케줄을 바탕으로 각 자율분산 역 제어시스템의 경로제어서브시스템에서 인접역에서 송신한 열차추적정보를 자동 비교하여 열차를 추적하고 경로제어 업무를 수행하게 된다.

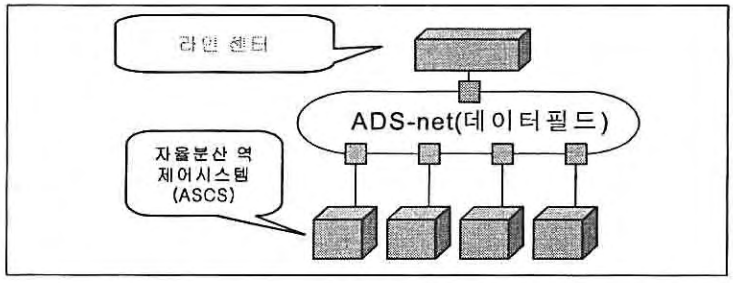


그림 3-79 자율분산 노선관리 확장 개념도

다. 라인정보 서브시스템의 도입에 따른 기존 CTC의 업무변화

라인정보 시스템은 국내 철도시스템에 도입되어 있지 않으므로 라인정보 시스템의 도입에 따라 기존 철도 관제센터의 업무 흐름 및 절차가 수정되어야 한다. 다음의 흐름도는 향후 라인센터 구축을 위한 계획의 일환으로 라인센터 도입에 따른 CTC 스케줄의 흐름도 변화에 대해 정리한 것이다.

(1) 기존 열차스케줄 관리 흐름도

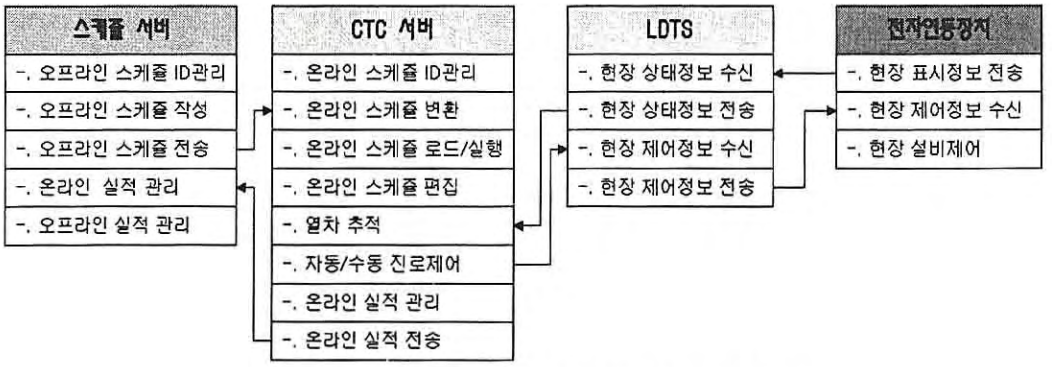


그림 3-80 기존 열차스케줄 관리 흐름도

기존 통합관제 CTC는 현장 설비와의 스케줄 전송에 대한 흐름이 없다. 이는 현장에서 역 단위의 스케줄을 관리하고 사용하는 시스템이 존재하지 않았기 때문이다. 또한 CTC와 LDTS간에도 스케줄 관련 프로토콜이 정의되어 있지 않다.

(2) 기존노선에 ADTMS가 도입되는 경우의 스케줄 관리 흐름도

현장에 ASCS가 설치되고 역 단위의 테스트가 완료 된 이후 라인정보 서브시스템이 구축되어 ASCS에 역 단위의 스케줄을 전송하게 된다. 기존 LDTS와 CTC와의 데이터 인터페이스는 동일하게 유지되며, ASCS는 LDTS에 연결되어 현장 설비에 대한 표시 및 제어를 수행하게 된다. 라인정보 서브시스템은 센터 설비에 새롭게 추가되어 설치되며, 라인정보 서브시스템은 스케줄서버와 CTC 서버에 연결된다.

스케줄 서버에서는 라인단위의 오프라인 스케줄에 대한 정보를 수집하여 라인정보 서브시스템이 관리하는 역에 역 단위 스케줄을 전송하며, CTC서버와 연결되어 당일 운행되고 있는 온라인 스케줄에 대한 정보를 수신하여 당일 운행계획에 대한 라인단위 스케줄을 관리하게 된다.

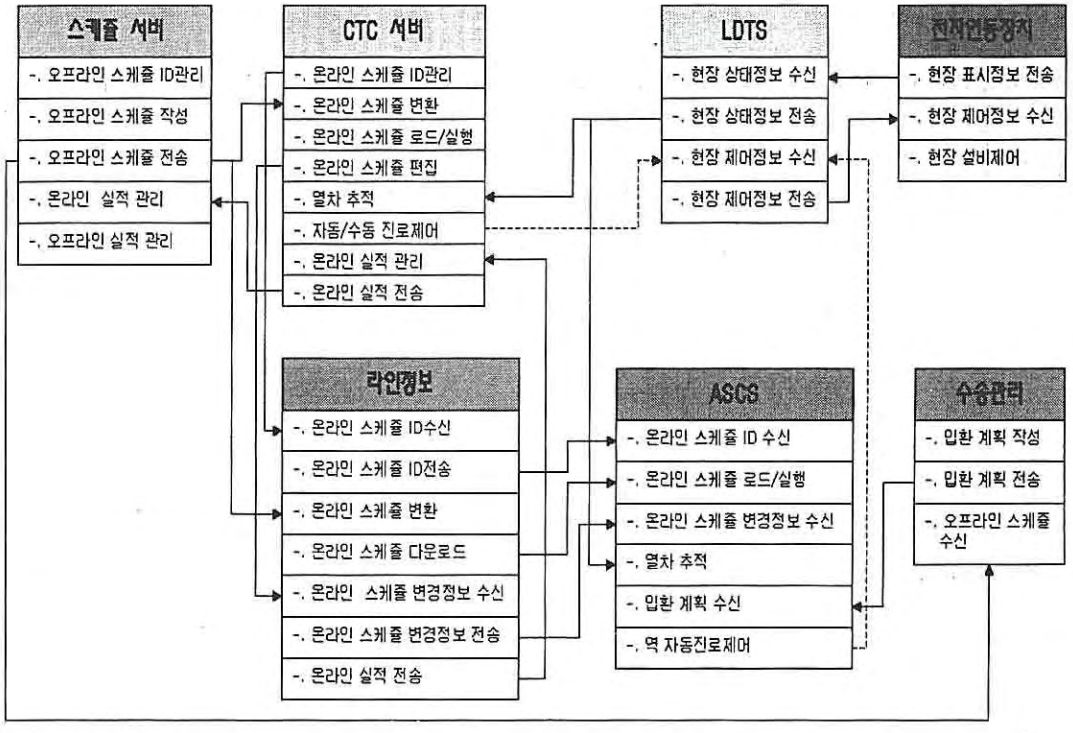


그림 3-81 기존노선에 ADTMS가 도입되는 경우의 스케줄 관리 흐름도

(3) 신규 노선에 ADTMS가 도입되는 경우의 스케줄 관리 흐름도

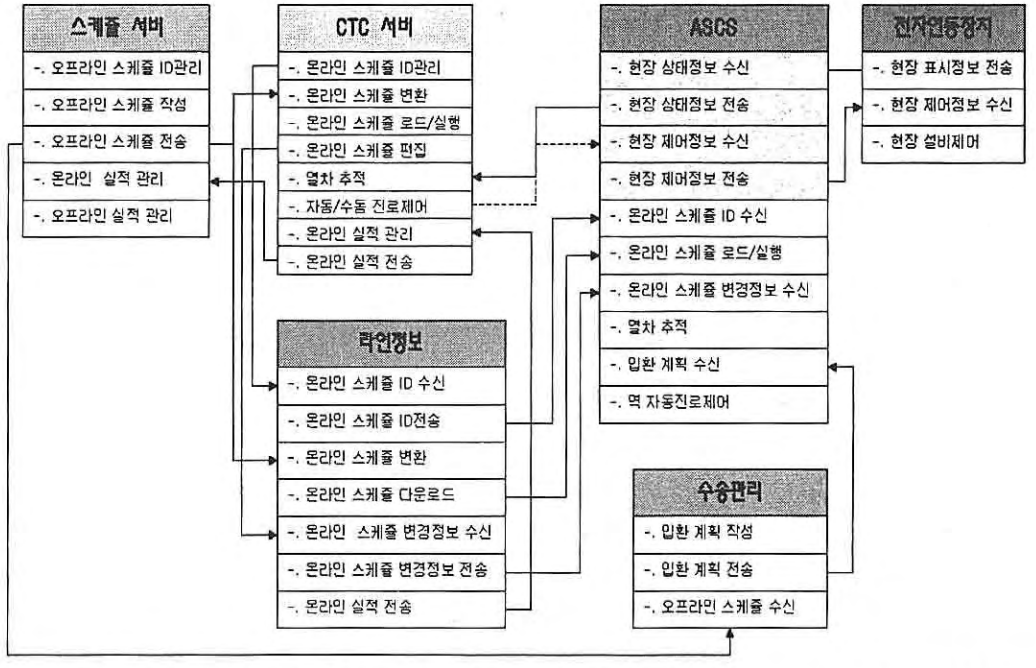


그림 3-82 신규 노선에 ADTMS가 도입되는 경우의 스케줄 관리 흐름도

신규 노선에 설치되는 ASCS는 기존 기능의 LDTS를 포함하도록 하는 경우와 기존 기능의 LDTS를 포함하지 않는 경우로 구분되며, 기존 기능의 LDTS를 포함하는 경우에는 기존역 ASCS 도입과 동일한 절차를 거치게 되며, 기존 LDTS를 포함하지 않는 경우에는 ASCS에 LDTS의 기능을 포함하도록 시스템의 설계 변경이 이루어 져야 한다. 스케줄 전송에 대한 라인정보 서브시스템의 기능은 앞의 단계와 동일하게 이루어진다.

(4) 스케줄 전송 흐름도

라인정보 서브시스템의 도입에 따른 스케줄 정보의 전송 흐름은 그림 3-83과 같다. 각 서브시스템간의 스케줄 전송은 크게 오프라인 스케줄 전송과 온라인 스케줄 전송으로 구분된다. 오프라인 스케줄은 일일 단위의 스케줄을 파일 형태로 전송하는 방식을 취하며, 온라인 스케줄은 각 시스템 간에 P2P로 연결되어 시스템에 로드되어 사용하고 있는 온라인 스케줄에 대한 변경 정보를 전송하는 경우에 사용된다

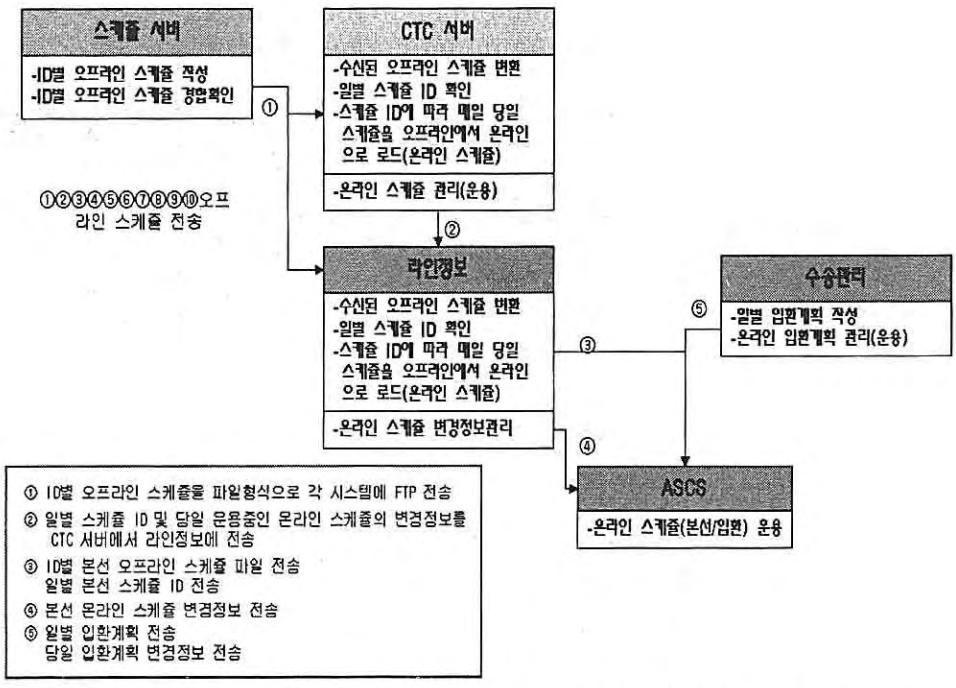


그림 3-83 라인정보 서브시스템의 스케줄 전송 흐름도

(5) 센터/역간의 데이터 동기화

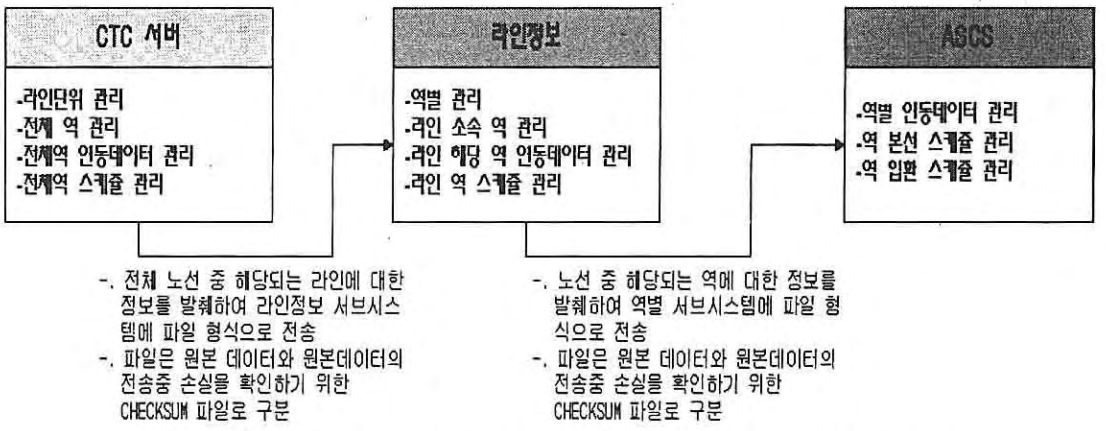


그림 3-84 라인정보 서브시스템과 ASCS간의 데이터 동기화

라인정보 서브시스템은 열차 운행관리를 위하여 열차 운행 관리에 필요한 기본 데이터를 역 ASCS와 동일하게 유지하여야 한다. 위의 그림3-84은 라인정보 서브시스템과 ASCS, 그리고 최종적으로는 CTC 서버와의 데이터 동기화에 대한 시퀀스를 나타낸 것이다.

(6) CTC센터와 라인센터의 스케줄 데이터베이스

열차스케줄은 기본으로 통합관제 스케줄 서버의 데이터베이스에 있는 전체 노선에 대한 데이터베이스 중에서 해당되는 노선의 데이터베이스만을 추출하고 내용을 재구성하여 라인단위의 스케줄을 재 생성하는 작업을 거쳐 데이터베이스를 새롭게 생성하는 것으로 한다.

차후 CTC와의 연계를 위하여 라인정보 서브시스템의 열차스케줄기능은 CTC서버의 온라인 열차운행계획에 대한 수정정보를 반영할 수 있도록 CTC서버와의 인터페이스가 정의되며, CTC서버에서의 온라인 데이터베이스 수정은 라인센터에서도 동일하게 반영될 수 있도록 CTC서버와 라인정보시스템에 기능이 구현된다. 라인정보 서브시스템의 운영자 MMI는 해당 라인에 대한 스케줄 관리 및 열차그래프 표시 기능이 구현된다.

라. 자율분산 시스템과의 병행운전 및 인터페이스 기능 구현

(1) ISS 개요

대형·고밀도 철도역 제어시스템의 구현을 위하여 기존의 통합관제 시스템 구성에 ASCS 및 라인정보 서브시스템이 새롭게 추가되며 자율분산 시스템을 위한 기존 CTC서버의 운영을 위해서는 ASCS역에 대한 정보 송수신을 위하여 새로운 인터페이스와 시스템이 정의된다.

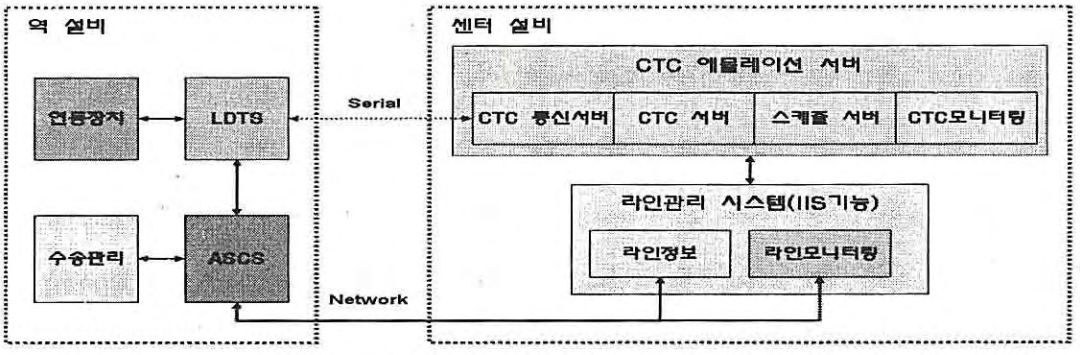


그림 3-85 자율분산시스템과의 병행운전을 위한
물리적 인터페이스 구성도

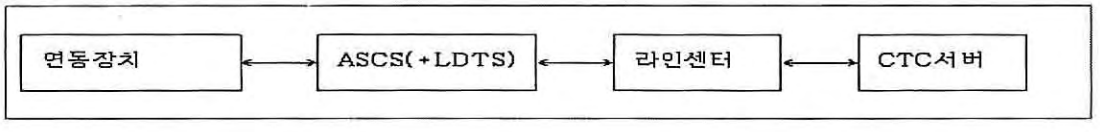
인터페이스 기능에는 다음과 같은 사항이 포함된다.

- 스케줄 서버의 오프라인 스케줄 (1일 단위의 오프라인 스케줄)
- 일별 스케줄 ID 설정정보
- 온라인 스케줄 변경정보

(2) ASCS 설치 환경 검토

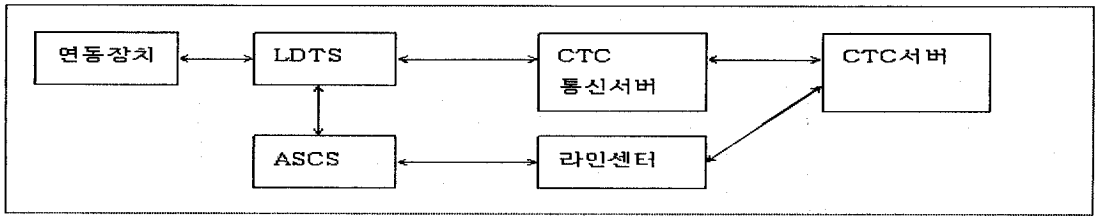
ASCS의 현장 설치는 다음과 같은 환경으로 구축한다.

(가) 신규 노선의 경우



신규 노선은 ASCS가 LDTS의 기능을 포함하도록 하여 현장 데이터 전달 절차를 최소화 되도록 하도록 한다. 또한 ASCS가 LDTS의 기능을 포함하므로 전체적인 장비 설치비용을 최소화할 수 있다.

(나) 기존 노선의 경우



기존 노선의 경우에는 기존 LDTS의 프로그램을 개선하여 CTC 통신서버와 라인센터에 동시에 데이터를 전송할 수 있도록 한다. 기존 CTC의 열차 모니터링은 연동장치 LDTS CTC통신서버 CTC서버 단계로 그대로 수행되며, 센터 스케줄을 CTC서버 라인센터 ASCS로 전달되어 ASCS가 역을 스케줄에 따라 자동 제어하도록 한다.

(3) ASCS 도입에 따른 CTC 서버/연동장치의 역 제어모드 변경

국내 철도시스템의 역 제어모드는 CTC-AUTO(스케줄 모드), CTC-MANUAL(센터운영자 수동제어), LOCAL(역운영자 수동제어) 모드로 구분된다.

연동장치의 역 제어모드는 LOCAL(역운영자 제어)과 CTC(센터 제어)로 구분된다.

ASCS가 도입되는 경우:

-. ASCS가 CTC 설비로 정의가 되면 센터 설비는 ASCS가 설치되어 있는 역의 경우에 역에서 자동으로 제어를 수행하도록 하기 위하여 CTC-ASCS모드를 추가하여야 한다.

-. ASCS가 역 설비로 정의가 되는 경우 역 제어모드는 LOCAL-ASCS모드, LOCAL-MANUAL(역 운영자 수동제어) 모드로 구분되어야 한다.

ASCS 시스템은 센터 설비인 경우에는 CTC-ASCS 모드인 경우에만 현장 설비에 대한 진로제어 명령을 출력하며, 역 설비인 경우에는 LOCAL-ASCS모드인 경

우에만 현장 설비에 대한 자동진로제어가 출력되도록 한다.

(4) 관제센터의 데이터 흐름 변화

대형·고밀도운행 철도역 자동운행제어시스템이 도입되면 구로 통합관제센터에 구축되어 있는 CTC센터설비의 데이터 흐름에도 변화가 필요하다.

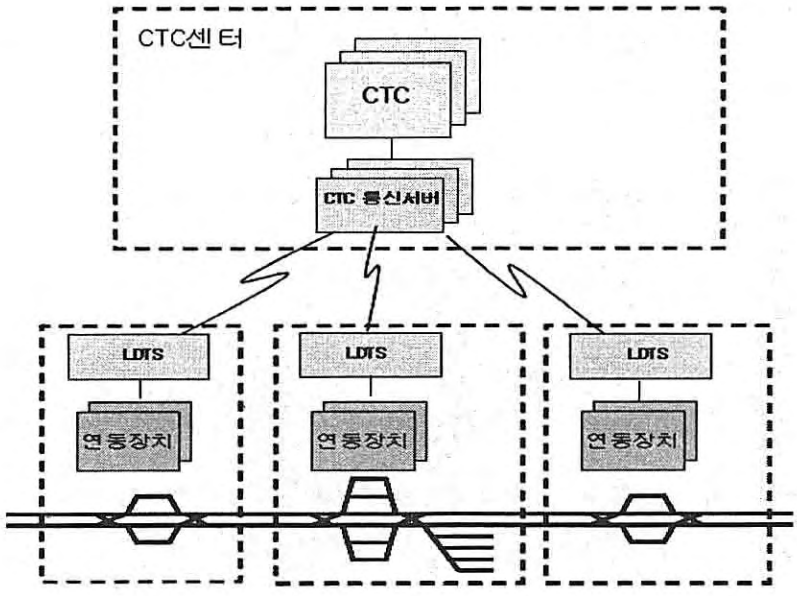


그림 3-86 기존 관제시스템의 연결 구조

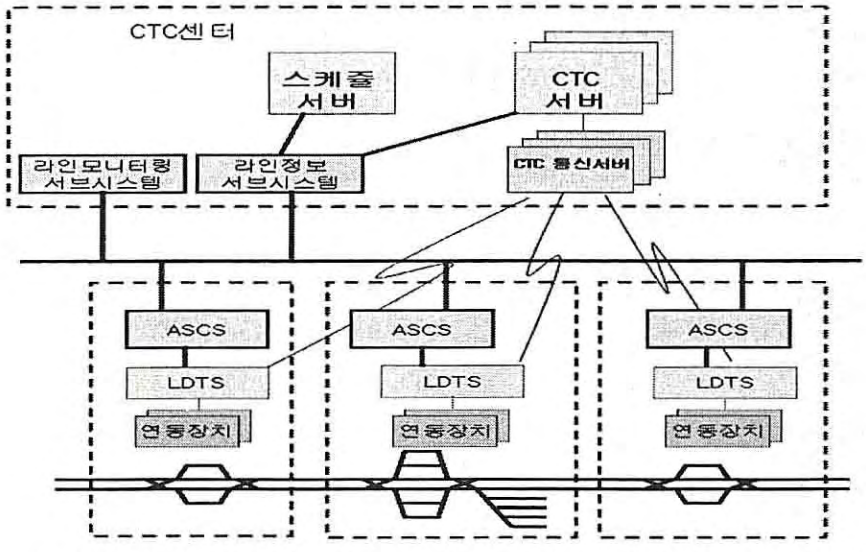


그림 3-87 ADTMS도입에 따른 관제시스템의 연결 구조 변화

현장역에 핵심서브시스템이 설치된 후 현장역에서 운용되는 본선구간에 대한 스케줄 관리를 위하여 센터에는 라인정보 서브시스템이 설치되어 스케줄관리 서버와 CTC서버에 연결되며 라인모니터링 서브시스템(5차년도)이 설치되어 ASCS를 통해 입력된 현장설비 상태정보와 LDTS를 통해 입력된 현장설비 상태정보의 표시상태를 비교할 수 있다.

(5) 라인센터와 역설비간의 Data Field 구성

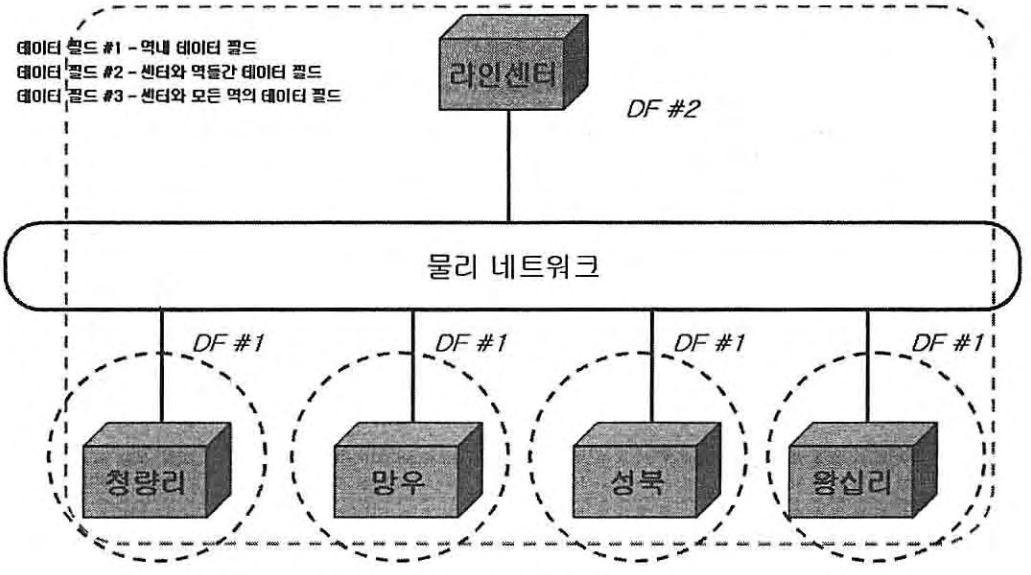


그림 3-88 ASCS와 라인센터의 Data Field 구성

라인센터 설비(라인정보 서브시스템, 라인모니터링 서브시스템)와 ASCS간의 데이터 입출력은 Data Field를 경유하여 이루어지며 기존 시스템인 CTC서버와 스케줄서버는 Data Field를 경유하지 않고 직접 P2P형태로 연결되어 오프라인 및 온라인 스케줄에 대한 정보를 관리한다.

(6) 라인센터와 CTC서버간의 인터페이스

라인정보 서브시스템의 기능 구현을 위하여 라인정보 서브시스템은 스케줄관리 서버와 연결되어 오프라인 스케줄을 수신하여 역 단위로 오프라인 스케줄을 전송하며, CTC서버와 연결되어 온라인 스케줄 변경정보를 수신하여 해당역에 대한 온라인 스케줄 변경정보를 송신하게 된다.

그림 3-89는 라인센터와 CTC서버, 스케줄 관리 서버간의 데이터 입출력에 대한 구성도를 나타낸 것이다.

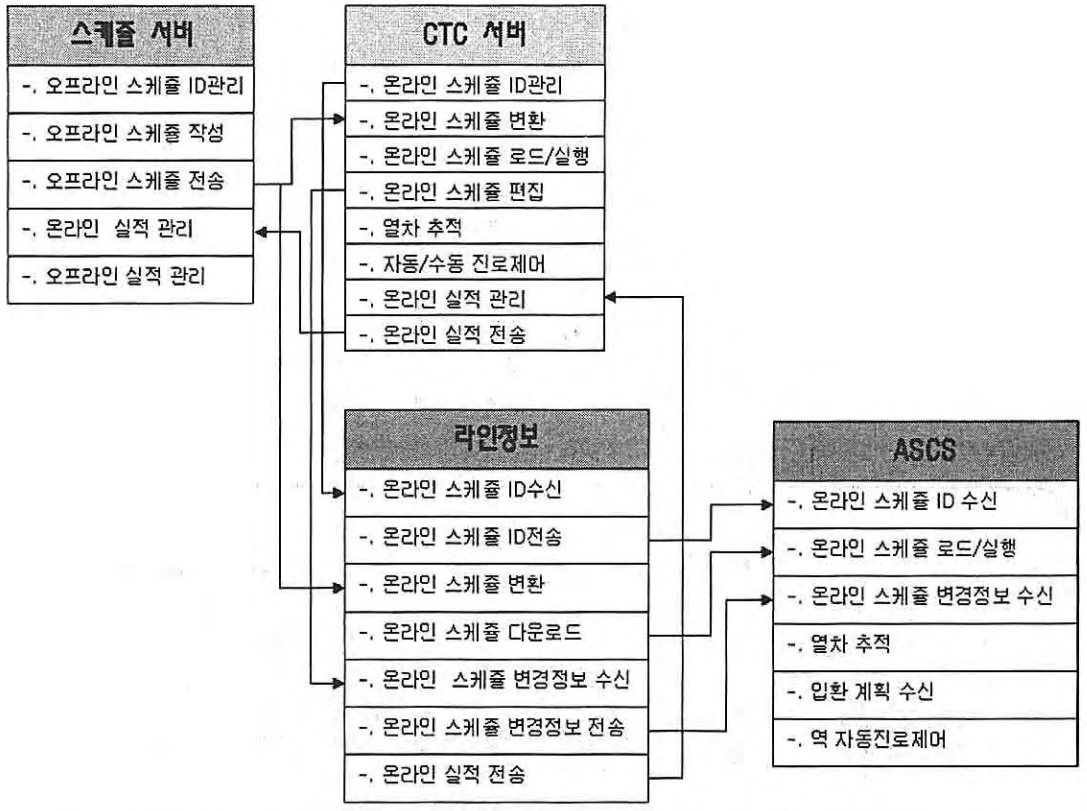


그림 3-89 라인정보 서브시스템과 CTC간의 데이터 흐름(기능측면)

스케줄 서버는 오프라인 스케줄이 변경되는 경우 CTC서버 및 라인정보 서브시스템에 오프라인 스케줄을 파일단위로 전송 후 오프라인 스케줄 전송이 완료되는 경우 각 시스템에 오프라인 스케줄 전송이 완료되었음을 통지하여 각 시스템의 기능이 수행되도록 한다. 라인정보 서브시스템은 파일단위의 오프라인 스케줄을 데이터베이스에 로드하여 역단위 스케줄 파일을 생성한 후 해당역에 역단위 스케줄을 파일로 전송한 후 전송 완료 정보를 역 ASCS시스템에 전송하여 스케줄 처리를 수행하도록 한다.

CTC서버는 당일 운행 예정인 온라인 스케줄에 대한 변경정보가 발생된 경우 라인정보 서브시스템에 변경되는 스케줄 정보를 전송하며, 라인정보 서브시스템은 수신된 스케줄 변경정보를 내부 처리 후 해당역에 역단위 스케줄 변경정보를 전송한다.

마. 업무흐름 파악

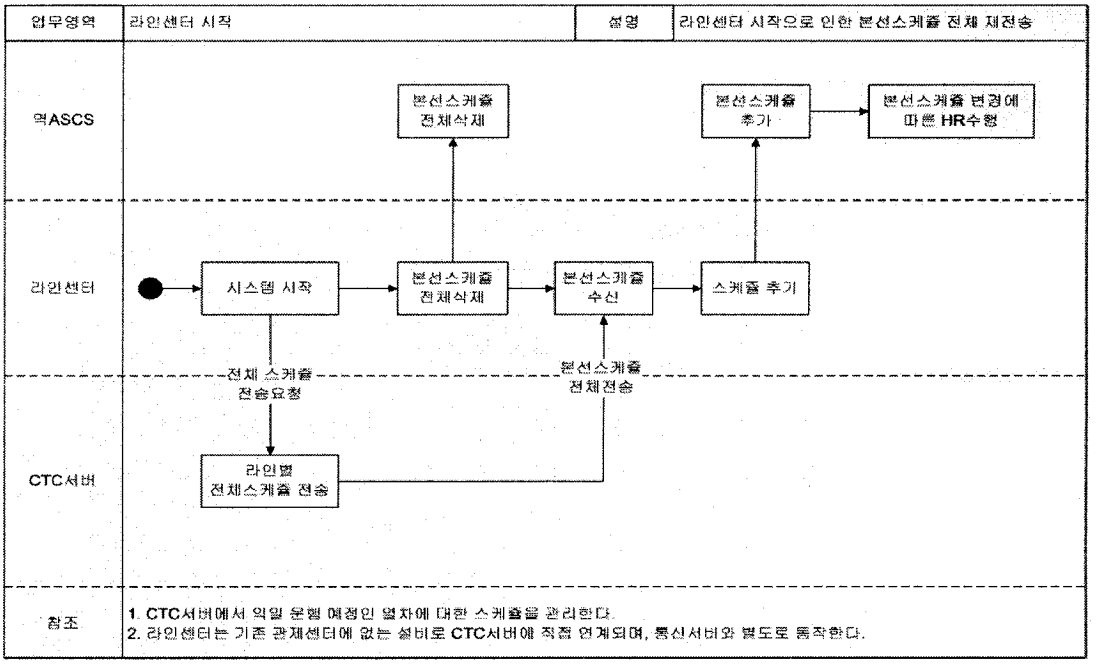


그림 3-90 라인센터 시작

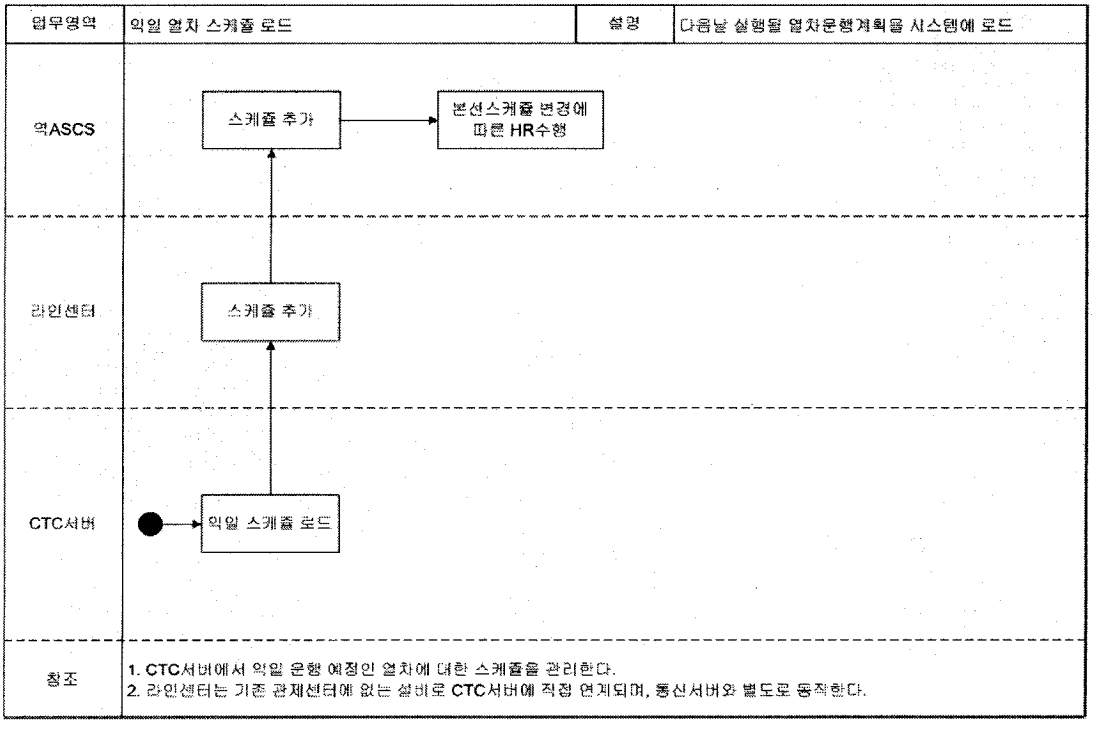


그림 3-91 익일 열차스케줄 로드

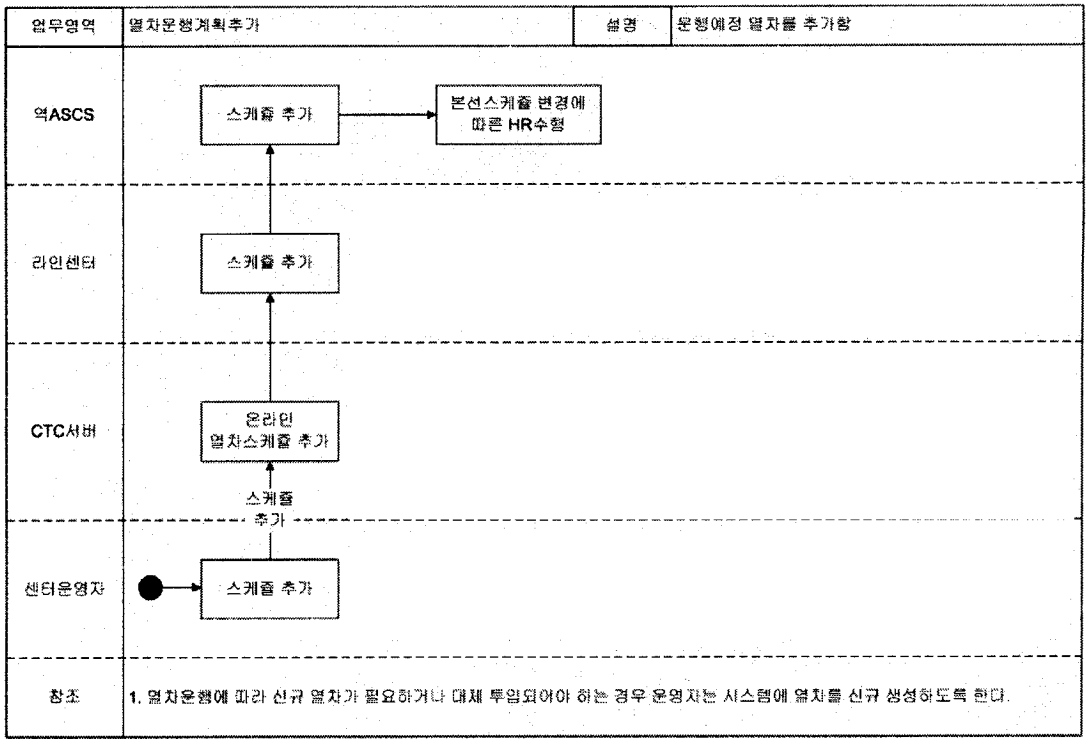


그림 3-92 열차운행계획 추가

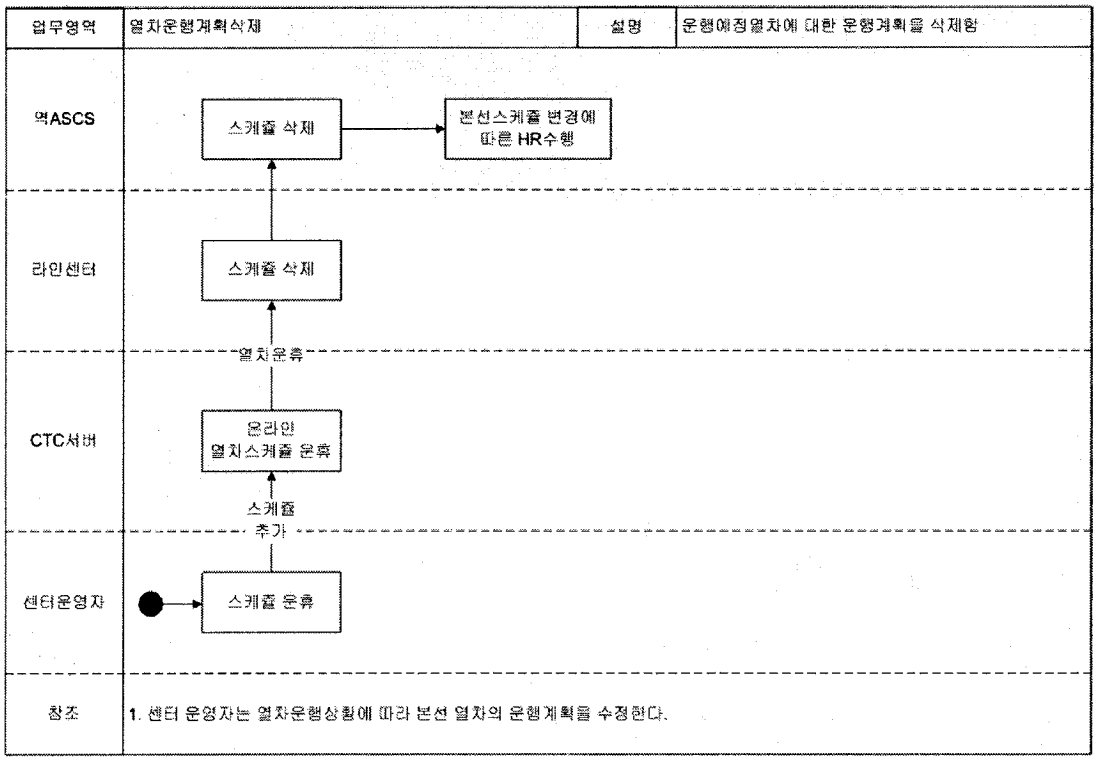


그림 3-93 열차운행계획 삭제

바. 유즈케이스 명세서

(1) 유즈케이스 명세서 : <변경스케줄 전송처리>

(가) 열차 목록 관리

① 요약설명

이 유즈케이스는 LMS 가 동작하는 환경에서 운영자에 의해 해당 라인의 온라인 본선스케줄이 추가/삭제/수정 등의 상황이 발생하는 경우 <ASCS>와의 데이터 동기화를 위해 역에 변경되는 스케줄 정보를 전송하는 유즈케이스이다.

(나) 이벤트 흐름

① 기본흐름 2 센터스케줄 변경 이벤트 수신 스케줄 추가

② <시스템>은 <CTC센터>로부터 수신된 본선스케줄 DB에 추가한다.

③ <시스템>은 추가된 본선스케줄의 해당역에 열차스케줄 추가를 전송한다.

④ 기본흐름 3 센터스케줄 변경 이벤트 수신 스케줄 수정

⑤ <중앙운영자>는 <CTC센터>의 GUI로 본선스케줄을 수정한다.

⑥ <시스템>은 <ADM>을 경유하여 <CTC센터>의 수정된 본선스케줄을 수신한다. 수신한 스케줄은 DB에 업데이트된다.<대체흐름 2.4.1 : 수정 요청된 열차스케줄이 라인센터 데이터베이스에 없는 경우>

⑦ <시스템>은 업데이트된 본선스케줄의 해당역에 열차스케줄 업데이트를 전송한다.

⑧ 기본흐름 4 센터스케줄 변경 이벤트 수신 스케줄 삭제

⑨ <중앙운영자>는 <CTC센터>의 GUI로 본선스케줄을 삭제한다.

⑩ <시스템>은 <ADM>을 경유하여 삭제된 본선스케줄을 수신한다. 수신한 스케줄은 DB에서 삭제된다.

⑪ <시스템>은 업데이트된 본선스케줄의 해당역에 열차스케줄 삭제를 전송한다.

⑫ 대체흐름

⑬ 수정 요청된 스케줄이 라인센터 데이터베이스에 없는 경우

- <시스템>은 해당 열차를 라인센터에 신규 등록하고 <ASCS>에 스케줄

추가를 수행한다.

- <시스템>은 수신된 열차번호가 <시스템>에 등록되지 않음을 운영자에게 알리고 로그로 저장한다.

(다) 사전조건

① 필요한 정보가 모두 초기화되고, 로드 또는 수신되어 유효해야 한다.

② 해당역의 현장 상태 정보, 스케줄 정보를 참조할 수 있어야 한다.

③ ADM이 구동되어 있어야 한다.

④ 다른 서브시스템과의 송수신을 위해ADM가 구동되어 있어야 한다.

⑤ 역의 운영모드가 적절히 관리되어야 한다.

⑥ 라인센터의 스케줄 전송 및 실적 수신을 위해서는 역의ASCS가 센터와 연결되어 있어야 한다. 역 운영모드가 역 단독모드라고 하더라도 센터와 접속이 이루어져 있는 경우에는 스케줄 다운로드가 가능한 경우로 간주하여 처리한다.

(라) 사후조건

스케줄과 실적은 <시스템>의 내용과 역 <ASCS>간에 항상 동기화 되도록 한다.

(2) 유즈케이스 명세서 : <당일스케줄 로드>

(가) 시스템 초기화

① 요약설명

이 유즈케이스는 LMS(Line Management System)가 시스템이 초기화 되었거나 내일신규로 운행이 예정되어 있는 열차에 대한 열차운행계획을 시스템의 운행 데이터베이스에 로드하여 라인센터가 운행스케줄 활용할 수 있도록 하는 유즈케이스이다.

(나) 이벤트 흐름

① 기본흐름 1 데이터베이스 초기화

② <시스템>은 라인센터의 스케줄 데이터 상태를 "운영준비" 모드로 설정한다.

③ <시스템>은 라인센터가 관리하는 DB의 당일스케줄을 모두 삭제한다.

④ <시스템>은 <ADM>을 경유하여 라인센터에 속한전체 역에 대해 기존스케줄 삭제 명령을 전송한다.

⑤ 기본흐름 2 당일스케줄 데이터베이스에 로드

⑥ <시스템>은 당일 열차스케줄을 파일에서 로드한다.

⑦ <시스템>은 로드된 파일을 데이터베이스에서 활용 가능한 데이터로 변환하여 데이터베이스에 당일 스케줄로 추가한다.

⑧ <시스템>은 데이터베이스에 로드된 스케줄에 대해서 전역으로 스케줄 전송 미완료 상태로 한다.

⑨ <시스템>은 라인센터의 스케줄 데이터 상태를 "운영가능" 모드로 설정한다.

⑩ <시스템>은 <ADM>을 경유하여 라인센터에 속한 전체 역에 대해 스케줄 전송이 준비되었음을 전송한다.

⑪ 기본흐름 3 내일스케줄 데이터베이스에 로드

⑫ <시스템>은 일정 시점에 내일 열차스케줄을 파일에서 로드한다.

⑬ <시스템>은 로드된 파일을 데이터베이스에서 활용 가능한 데이터로 변환하여 데이터베이스에 내일 스케줄로 추가한다.

⑭ 기본흐름 4 당일스케줄 데이터베이스에 재로드

⑮ <시스템>은 당일 열차스케줄을 파일에서 로드한다.

⑯ <시스템>은 재 로드할 당일 열차스케줄에 대해서 임시 테이블을 생성하고 임시 테이블에 파일에서 가져온 데이터를 데이터베이스에서 활용 가능한 데이터로 변환하여 추가한다.

⑰ <시스템>은 임시테이블에 이미 당일 열차운행계획에 운행이 완료된 구간이나 역에 대해서 업데이트를 수행한다.

⑱ <시스템>은 임시테이블에 운행이 안 된 구간의 스케줄에 대해서 스케줄 미전송상태로 업데이트를 수행한다.

⑲ <시스템>은 스케줄 데이터 상태를 "운영준비" 모드로 설정한다.

⑳ <시스템>은 <ADM>을 경유하여 라인센터에 속한전체 역에 대해 기존스케줄 삭제 명령을 전송한다.

① <시스템>은 당일열차스케줄에 대한 테이블을 새로운 임시테이블로 명칭을 변경한다.

② <시스템>은 라인센터의 스케줄 데이터 상태를 "운영가능" 모드로 설정한

다.

③ 대체흐름

④ 열차운행이 완료된 시점의 열차운행계획을 시스템에 로드하는 경우

- <시스템>은 운행 완료된 역에 대해서는 역으로 해당 열차가 운행 완료된 것으로 간주하여 역으로 스케줄을 다운로드 하지 않는다.

⑤ 열차운행중인 시점의 열차운행계획을 시스템에 로드하는 경우

- <시스템>은 운행 완료된 역에 대해서는 역으로 해당 열차가 운행 완료된 것으로 간주하여 역으로 스케줄을 다운로드 하지 않는다.

(다) 사전조건

① 필요한 정보가 모두 초기화되고, 로드 또는 수신되어 유효해야 한다.

② 해당 <시스템>의 시스템 상태정보, 스케줄 정보를 참조할 수 있어야 한다.

③ ADM이 구동되어 있어야 한다.

④ 다른 서브시스템과의 송수신을 위해ADM이 구동되어 있어야 한다.

⑤ 역의 운영모드가 적절히 관리되어야 한다.

⑥ 스케줄 다운로드가 가능한 역의운영모드가 센터 연동모드로 구동 중이어야 한다.

(라) 사후조건

스케줄은 전체 역에 즉시 송신하여 동기화되도록 한다.

(3) 유즈케이스 명세서 : <전체스케줄 전송처리>

(가) 열차 목록 관리

① 요약설명

이 유즈케이스는 LMS 가 처음 기동되었거나 재로드되어 역단위의 전체 스케줄을 전송하는 유즈케이스이다.

(나) 이벤트 흐름

① 기본흐름 1 전체스케줄 전송요청 수신

② <시스템>은 <ASCS>로부터 <ADM>을 경유하여 전체 본선스케줄 전송요청 수신한다.

③ <시스템>은 해당역에 대한 전체스케줄 전송을 처리하는 쓰레드를 생성한다.

④ 기본흐름 2 전체스케줄 수신완료 수신

⑤ <시스템>은 <ASCS >로부터 수신된 해당역 본선스케줄 수신이 완료되었음을 수신한다.

⑥ <시스템>은 해당역의 본선스케줄에 대해송신 상태를 완료로 업데이트 한다.

⑦ <시스템>은 송신 완료한 열차정보에 대해로그로 기록한다.

⑧ <시스템>은 해당역에 대한 전체스케줄 전송이 완료된 경우 해당 역의쓰레드를 종료한다.

⑨ 기본흐름 3 전체스케줄 수신완료 미수신

⑩ <시스템>은 해당역에 대한 전체스케줄 전송을 처리하는 쓰레드를 종료한다.

⑪ 기본흐름 4 역 전체스케줄 전송 쓰레드 생성

⑫ <시스템>은 <ASCS>를 위하여 해당역의 당일 미전송 스케줄에 대해서 최대 크기를 가지는 패킷을 생성한다.

⑬ <시스템>은 <ADM>을 경유하여 <ASCS>에 전체 패킷을 순서대로 전송한다.

⑭ 각 패킷의 전송 간격은 100ms 이상으로 하며 전송 완료 후 100ms 이내에 해당 패킷 수신완료가 설정되지 않은 경우에는 3회에 한하여 재전송한다.

⑮ 대체흐름

⑯ 지정된 시간 이내에 전체 스케줄 전송 완료가 안 되는 경우

- <시스템>은 송신 완료되지 않은 패킷에 대해서 스케줄 전송 에러를 기록한다.
- <시스템>은 송신 완료되지 않은 열차번호에 대해서 수동으로 스케줄 전송을 수행하도록 요청한다.

(다) 사전조건

① 필요한 정보가 모두 초기화되고, 로드 또는 수신되어 유효해야 한다.

해당역의 현장 상태 정보, 스케줄 정보를 참조할 수 있어야 한다.

② ADM이 구동되어 있어야 한다.

다른 서브시스템과의 송수신을 위해 ADM이 구동되어 있어야 한다.

③ 역의 운영모드가 적절히 관리되어야 한다.

라인센터의 스케줄 전송 및 실적 수신을 위해서는 역의ASCS가 센터와 연결

되어 있어야 한다. 역 운영모드가 역 단독모드라고 하더라도 센터와 접속이 이루어져 있는 경우에는 스케줄 다운로드가 가능한 경우로 간주하여 처리한다.

(라) 사후조건

스케줄과 실적은 <시스템>의 내용과 역 <ASCS>간에 항상 동기화 되도록 한다.

(4) 유즈케이스 명세서 : <실적 수신 처리>

(가) 열차 목록 관리

① 요약설명

이 유즈케이스는 LMS 가 역 ASCS에서 현재 운행 중인 열차에 대한 역 실적을 수신하여 처리하는 유즈케이스이다.

(나) 이벤트 흐름

① 기본흐름 1 역실적 일괄 수신

② <시스템>은 <ASCS>로부터 <ADM>을 경유하여 역실적을 수신한다.

③ <시스템>은 수신한 실적은 DB에 업데이트된다. <대체 흐름2.3.1 스케줄상에 수신한 실적의 열차번호가 존재하지 않는 경우>

④ <시스템>은 수신된 실적이 종착역 종료실적(종료 후 입환)이면 해당 열차에 대해서는 당일 열차 운행이 종료된 것으로 처리한다.

⑤ 기본흐름 2 - <ASCS>로부터의 실적 이벤트 수신

⑥ <시스템>은 열차 착발이 발생할 때마다 <ASCS>로부터 <ADM>을 경유하여 실적 이벤트를 수신한다. 수신한 실적은 DB에 업데이트된다. <대체 흐름 2.3.1 스케줄상에 수신한 실적의 열차번호가 존재하지 않는 경우>

⑦ <시스템>은 수신된 실적이 종착역 종료실적(종료 후 입환)이면 해당 열차에 대해서는 당일 열차 운행이 종료된 것으로 처리한다.

⑧ 대체흐름

⑨ 스케줄상에 수신한 실적의 열차번호가 존재하지 않는 경우

- <시스템>은 "스케줄상에 존재하지 않는 열차번호 실적 수신"이라는 에러를 표시하고 로깅하고 해당 수신 실적을 DB에 추가한다.

(다) 사전조건

① 필요한 정보가 모두 초기화되고, 로드 또는 수신되어 유효해야 한다.

② 해당역의 현장 상태 정보, 스케줄 정보를 참조할 수 있어야 한다.

③ ADM이 구동되어 있어야 한다.

④ 다른 서브시스템과의 송수신을 위해 ADM이 구동되어 있어야 한다.

⑤ 역의 운영모드가 적절히 관리되어야 한다.

⑥ 라인센터의 스케줄 전송 및 실적 수신을 위해서는 역의 ASCS가 센터와 연결되어 있어야 한다. 역 운영모드가 역 단독모드라고 하더라도 센터와 접속이 이루어져 있는 경우에는 스케줄 다운로드가 가능한 경우로 간주하여 처리한다.

(라) 사후조건

스케줄과 실적은 <시스템>의 내용과 역 <ASCS>간에 항상 동기화 되도록 한다.

(5) 유즈케이스 명세서 : <스케줄 정보 관리>

(가) 유즈케이스명

① 요약설명

이 유즈케이스는 <LMS>가 스케줄 정보를 입력, 수정, 삭제 하는 처리하는 유즈케이스이다.

(나) 이벤트 흐름

① 기본흐름(스케줄 입력)

② <운영원>은 MMI화면에서 스케줄 입력 메뉴를 선택한다.

③ <시스템>은 스케줄 입력화면에 노선전체 스케줄 정보를 보여준다.

④ <운영원>은 스케줄 입력화면에서 열차정보를 입력한다.(열차번호)

⑤ <운영원>은 스케줄 입력화면에서 <시작역>을 입력한다.

⑥ <시스템>은 스케줄 입력화면에서 <시작역>의 진입신호기 목록을 표시한다.

⑦ <운영원>은 스케줄 입력화면에서 <진입신호기>를 선택한다.

⑧ <운영원>은 스케줄 입력화면에서 <종착역>을 입력한다.

⑨ <운영원>은 스케줄 입력화면에서 <진입시간>을 입력한다.

⑩ <운영원>은 스케줄 입력화면에서 <스케줄생성> 버튼을 누른다.

⑪ <시스템>은 스케줄 입력화면에서 <시작역>부터 <종착역>까지 역별 스케줄을 생성하여 표시한다.

⑫ <운영원>은 해당 열차의 역 목록에서 편집을 수행할 역을 선택한다.

⑬ <시스템>은 스케줄 입력화면에 해당 역 스케줄 정보를 보여준다.

⑭ <운영원>은 스케줄 입력화면에서 역 정보를 입력한다.(도착 진로, 출발진로, 도착시간, 출발시간)

⑮ <운영원> 업데이트 버튼을 누른다.

⑯ <운영원>은 해당 열차번호의 전체 역에 대해서 11 - 14번 과정을 반복한다.

⑰ <운영원> 스케줄 생성 완료 버튼을 누른다.

⑱ <시스템>은 생성된 본선스케줄 변경 메시지를 <역 ASCS>로 보낸다.

⑲ <시스템>은 스케줄 입력화면에 해당 역 스케줄을 다시 표시한다.

⑳ 2.2 기본흐름(스케줄 수정)

① <운영원>은 MMI화면에서 스케줄 수정/삭제 메뉴를 선택한다.

② <시스템>은 스케줄 수정/삭제 화면에서 열차번호를 선택한다.

③ <시스템>은 스케줄 수정/삭제 화면에서 해당 열차의 전체 스케줄 정보를 보여준다.

④ <운영원>은 스케줄 수정/삭제 화면에서 수정을 원하는 열차번호를 선택한다.

⑤ <시스템>은 선택된 스케줄 정보를 화면 하단에 수정할 수 있도록 표출한다.

⑥ <운영원>은 스케줄 수정/삭제 화면에서 스케줄 정보를 찾아서 수정한다.<대체흐름 2.4.1 : 스케줄 수정시 이미 운행 완료된 스케줄을 수정하는 경우>

⑦ <운영원>은 스케줄 수정작업이 끝나면 수정완료 버튼을 누른다.

⑧ <시스템>은 수정된 본선스케줄 변경 메시지를 <역 ASCS>로 보낸다.

⑨ 기본흐름(스케줄 삭제)

⑩ <운영원>은 MMI화면에서 스케줄 수정/삭제 메뉴를 선택한다.

⑪ <시스템>은 스케줄 수정/삭제화면에 해당 역 스케줄 정보를 보여준다.

⑫ <운영원>은 스케줄 수정/삭제 화면에서 삭제를 원하는 열차번호를 선택한다.

⑬ <운영원>은 삭제 버튼을 누른다.<대체흐름 2.4.2 : 스케줄 삭제시 현재운행

중이 스케줄을 삭제하는 경우>

⑭ <시스템>은 삭제된 본선스케줄 변경 메시지를 <역 ASCS>로 보낸다.

⑮ 대체흐름

⑯ 스케줄 수정시 이미 운행 완료된 스케줄을 수정하는 경우

<시스템>은 운영자가 이미 운행이 종료된 구간에 대한 스케줄을 수정하는 경우 해당 스케줄이 이미 종료되었으므로 수정이 불가능함을 알린다.

⑰ 스케줄 삭제시 현재운행중이 스케줄을 삭제하는 경우

<시스템>은 운영자가 현재 열차가 운행 중임을 <운영원>에게 표시하고 계속 진행할 것인가를 확인한다. 만일 운영자가 계속 진행하기를 원하는 경우나머지 운행예정 구간에 대해서 스케줄 삭제를 처리한다.

(다) 사전조건

<ASCS> 시스템이 정상적으로 가동 되어 있어야 한다.

(6) 유즈케이스 명세서 : <센터스케줄 수신>

(가) 유즈케이스명

① 요약설명

이 유즈케이스는 LMS(Line Management System) 가 날짜별로 설정된 본선 구간 스케줄을 수신하여 일별 열차 운행계획 로드가 가능한 형태로 저장하는 유즈케이스이다.

(나) 이벤트 흐름

① 기본흐름 1 본선스케줄 헤더 수신

② <시스템>은 <CTC 시스템>에서 수신된 헤더별 종류에 따른 신규 스케줄 파일을 생성한다. 만일 헤더 패킷이 비정상인 경우에는 신규 스케줄 파일을 생성하지 않고 바로 헤더 에러를 <CTC 시스템>에 전송하여 처음부터 다시 전송하도록 한다.

③ 기본흐름 2 본선스케줄 본문 수신

④ <시스템>은 <CTC 시스템>에서 본선구간 스케줄을 수신하여 생성된 스케줄 파일의 지정된 위치에 기록한다. <대체흐름 2.5.2:패킷별 검사결과가 정상

이지 않은 경우 >

⑤ 기본흐름 3 본선스케줄 검사 수신

⑥ <시스템>은 <CTC 시스템>에서 수신된 본선구간 스케줄이 헤더정보에 기록된 모든 정보가 다 수신완료 되었는지를 검사하여 <CTC 시스템>에 미수신된 정보에 대한 재전송을 요청한다.

⑦ <시스템>은 본선 구간 스케줄 수신이 완료된 경우 검사 정상완료를 전송한다.

⑧ 기본흐름 4 본선스케줄 종료 수신

⑨ <시스템>은 <CTC 시스템>에서 수신된 본선구간 스케줄 파일을 닫는다. <대체흐름 2.5.1:검사가 완료되지 않은 경우에 본선스케줄 종료가 수신된 경우>

⑩ <시스템>은 수신된 본선구간 스케줄 파일에서 해당 라인센터가 관리하는 역의 스케줄에 대해서 <시스템>에 로드가 가능하도록 해당라인의 본선구간에 대한 스케줄 파일을 생성한다.

⑪ <시스템>은 라인센터용 스케줄 파일이 작성 완료된 경우 본선구간 전체에 대한 스케줄 파일을 삭제한다.

⑫ 대체흐름

⑬ 검사가 완료되지 않은 경우에 본선스케줄 종료가 수신된 경우

- <시스템>은 검사가 완료되지 않은 상황에서 <CTC 시스템>에서 본선스케줄 전송 완료메시지가 수신된 경우 수신된 스케줄 파일을 삭제한 후 본선스케줄 <CTC 시스템>에 재전송 요청한다.

⑭ 패킷별 검사결과가 정상이지 않은 경우

- <시스템>은 수신된 패킷을 검사하여 비정상으로 판단된 패킷에 대해서는 파일에 저장하지 않으며 추후 검사 메시지가 수신되는 경우해당 패킷에 대한 재전송을 요청한다.

(다) 사전조건

① 필요한 정보가 모두 초기화 되고, 로드 또는 수신되어 유효해야 한다.

② 해당 <시스템>의 시스템 상태정보, 스케줄 정보를 참조할 수 있어야 한다.

③ ADM이 구동되어 있어야 한다.

④ 다른 서브시스템과의 송수신을 위해 ADM이 구동되어 있어야 한다.

⑤ 역의 운영모드가 적절히 관리되어야 한다.

⑥ 스케줄 다운로드가 가능한 역의운영모드가 센터 연동모드로 구동 중이어야 한다.

⑦ <CTC 시스템>과 라인센터의 연동데이터가 동기되어야 한다.

⑧ 스케줄 정보를 현장에서 적용가능하기 위해서는 <CTC 시스템> <라인센터> <ASCS>간의 연동데이터가 동기화 되어있어야 한다.

(라) 사후조건

스케줄은 전체 역에 즉시 송신하여 동기화되도록 한다.

사. 클래스 다이어그램

(1) 열차위치추적 표시 클래스 다이어그램

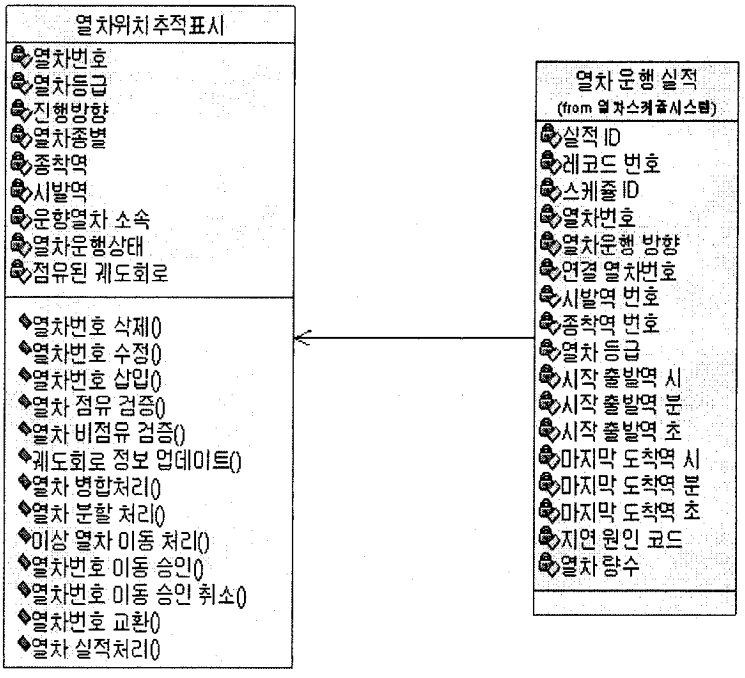


그림 3-94 열차위치추적 표시 클래스 다이어그램

(2) <오프라인 스케줄 작성> 클래스 다이어그램

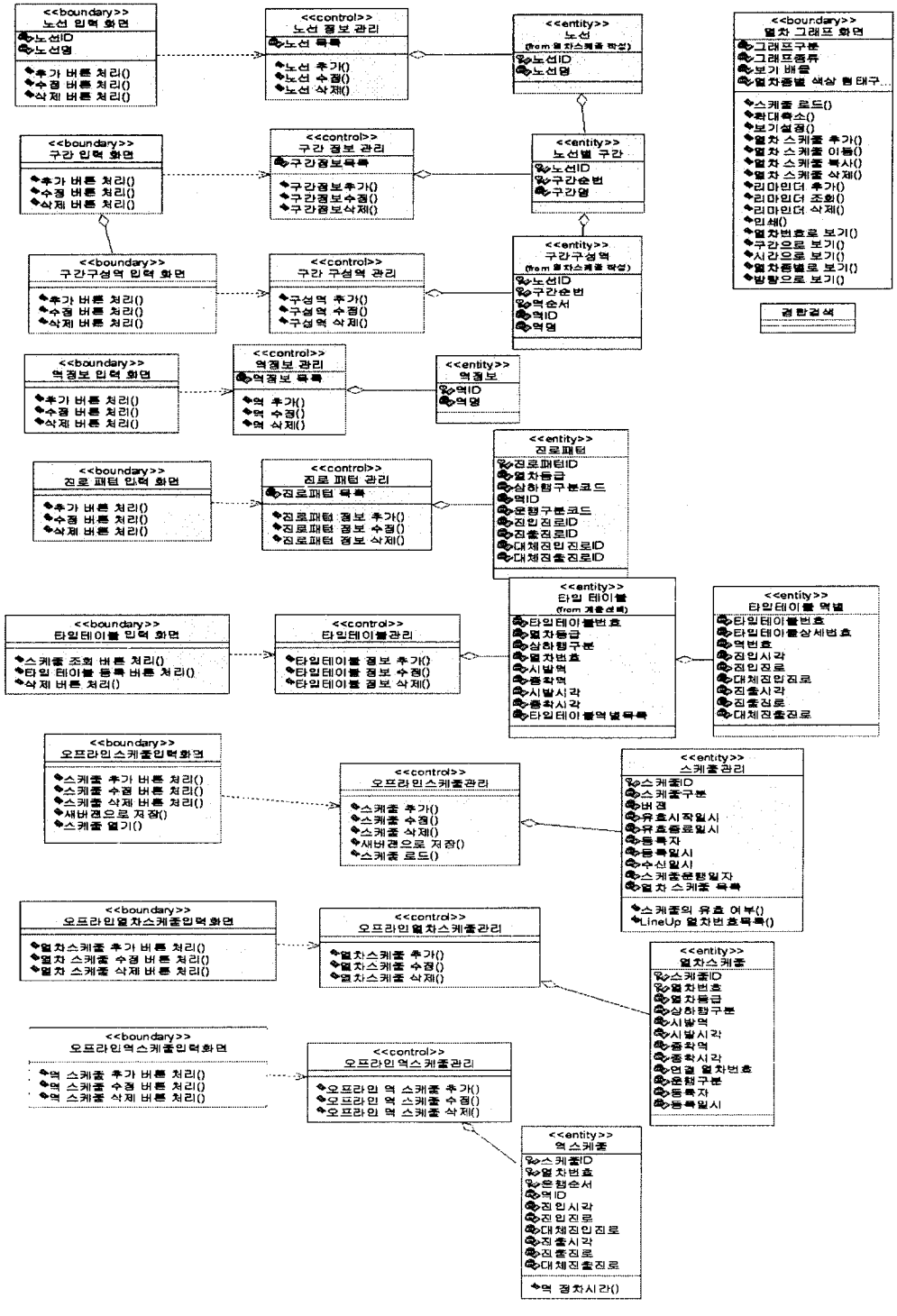


그림 3-95 < 오프라인 스케줄 작성 > 클래스 다이어그램

(3) 스케줄관리 클래스에 대한 속성 정의

표 3-41 스케줄관리 클래스에 대한 속성 정의

속성명	데이터 타입	속성정의
스케줄ID	String	하루치 열차스케줄에 대한Identification
스케줄구분	String	오프라인:0 온라인:1 외부수신 : 2
버전	String	스케줄구분별 버전
유효시작일시	Date	스케줄의 효력이 발생하기 시작하는 일시
유효종료일시	Date	스케줄의 효력이 만료되는 일시
등록자	String	스케줄의 최종작업 관제사의 ID
등록일시	Date	스케줄의 최종작업일시
수신일시	Date	외부 수신 스케줄의 경우 수신한 일시
스케줄운행일자	Date	외부 수신 스케줄, 오프라인 스케줄의 경우 스케줄을 사용하여 열차를 운행할 일자

(가) 열차스케줄 클래스에 대한 속성 정의

표 3-42 열차스케줄 클래스에 대한 속성 정의

속성명	데이터 타입	속성정의
스케줄ID	String	하루치 열차스케줄에 대한 Identification
열차번호	String	운행할 열차의 번호
열차등급	String	운행할 열차의 등급(새마을,무궁화...)
시발역	String	운행할 열차의 시발역번호
시발시각	Date	시발역 도착 시각
종착역	String	운행할 열차의 종착역번호
종착시각	Date	종착역 도착 시각
연결 열차 번호	String	연결할 열차번호
운행구분	Number	회송, 영업구분
등록자	String	최종 작업한 관제사 ID
등록일시	Date	최종 작업일시

(나) 역스케줄 클래스에 대한 속성 정의

표 3-43 역스케줄 클래스에 대한 속성 정의

속성명	데이터 타입	속성정의
스케줄ID	String	하루치 열차스케줄에 대한 Identification
열차번호	String	운행할 열차의 번호
운행순서	int	운행할 열차의 해당역에 대한 운행 순서
역ID	String	해당 역번호
진입시각	Date	해당역 도착 시각
진입진로	String	해당역 진입진로의 Element ID
진출시각	Date	해당역 출발 시각
진출진로	String	해당역 진출진로의 Element ID
대체진입진로	String	해당역 진입진로가 불가능할 때의 대체할 진로 Element ID
대체진출진로	String	해당역 진출진로가 불가능할 때의 대체할 진로 Element ID

(다) 진로패턴 클래스에 대한 속성 정의

표 3-44 진로패턴 클래스에 대한 속성 정의

속성명	데이터 타입	속성정의
진로패턴ID	String	진로 패턴에 대한 Identification
열차등급	String	운행할 열차의 등급
상하행구분코드	int	운행할 열차의 상하행구분
역ID	String	해당 역번호
운행구분코드	int	회송, 영업 구분
진입진로	String	해당역 진입진로의 Element ID
진출진로	대체진입진로	해당역 진출진로의 Element ID
대체진입진로	String	해당역 진입진로가 불가능할 때의 대체할 진로 Element ID
대체진출진로	String	해당역 진출진로가 불가능할 때의 대체할 진로 Element ID

(라) 타임 테이블 클래스에 대한 속성 정의

표 3-45 타임 테이블 클래스에 대한 속성 정의

속성명	데이터 타입	속성정의
타임테이블ID	String	타임 테이블에 대한 Identification
열차등급	String	운행할 열차의 등급
시발역	String	운행할 열차의 시발역
시발시각	Date	시발역 진출시각
종착역	String	운행할 열차의 종착역
종착시각	Date	종착역 진입시각
등록자	String	최종 등록자
등록일시	Date	최종 등록일

(마) 타임테이블 역별 클래스에 대한 속성 정의

표 3-46 타임테이블 역별 클래스에 대한 속성 정의

속성명	데이터 타입	속성정의
타임테이블ID	String	타임테이블에 대한 Identification
타임테이블 역별 ID	String	타임테이블 역별 정보에 대한 Identification
역ID	String	운행할 열차의 시발역
진입시각	Date	해당역 진입시각
진입진로	String	역 진입시 진로
대체진입진로	Date	진입진로가 불가능할 때 대체할 진로
진출시각	Date	해당역 진출시각
진출진로	String	역 진출시 진로
대체진출진로	Date	진출진로가 불가능할 때 대체할 진로
등록자	String	최종 등록자
등록일시	Date	최종 등록일시

(바) <열차스케줄 수신> 클래스 다이어그램

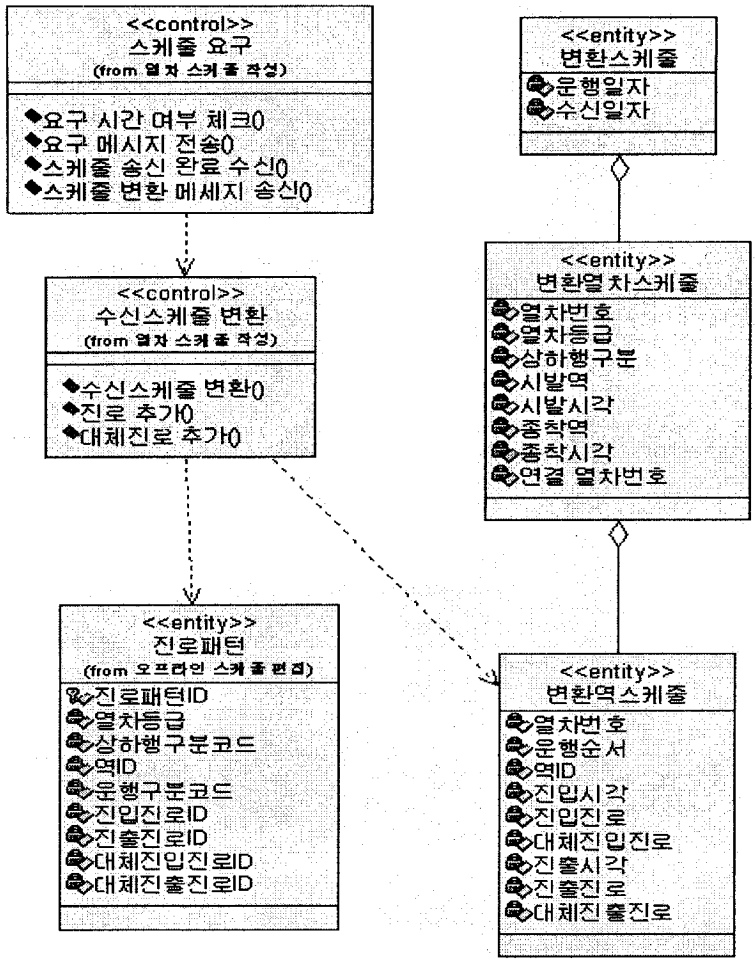


그림 3-96 열차스케줄 수신 클래스 다이어그램

(사) <온라인 스케줄 수신> 클래스 다이어그램

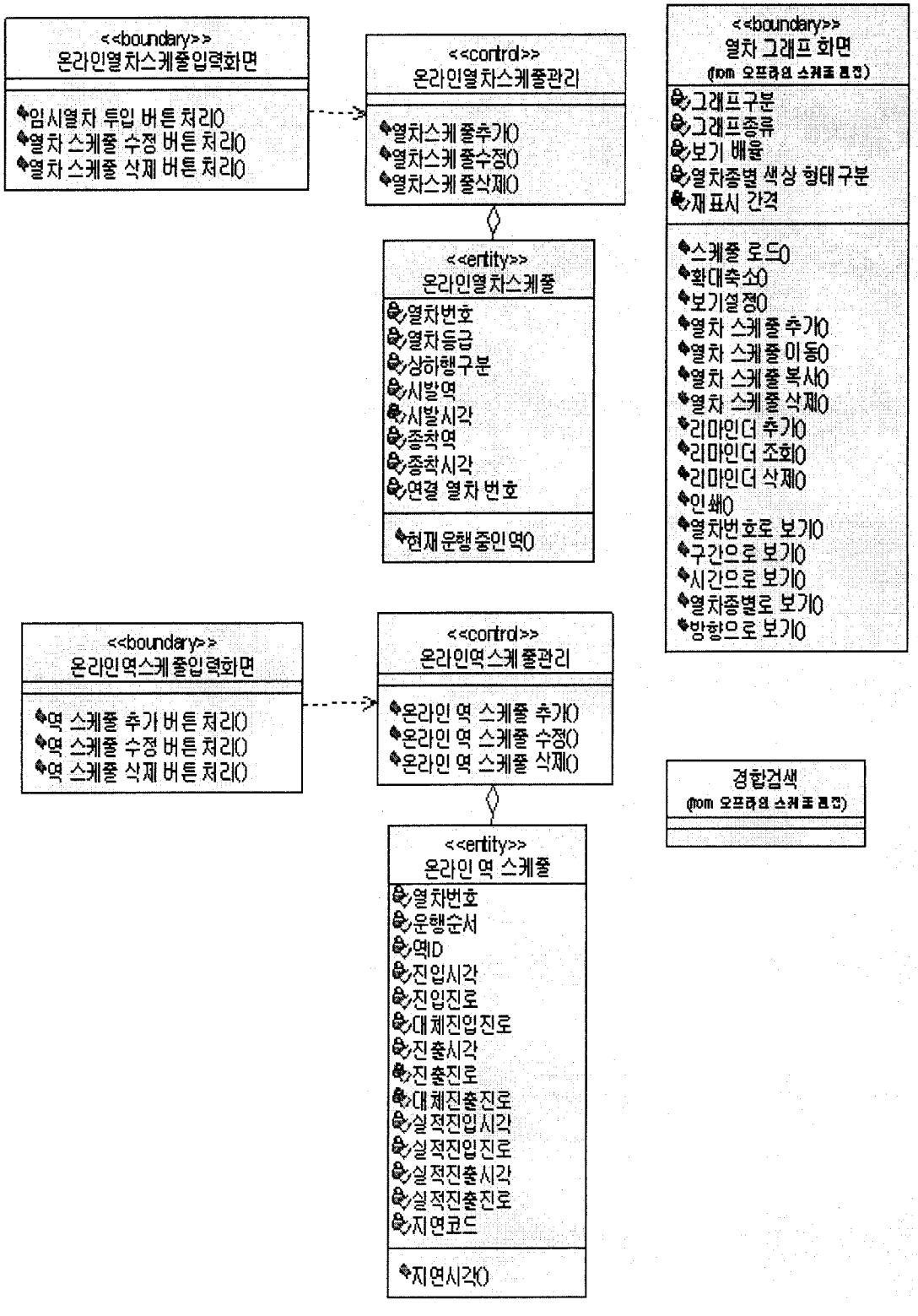


그림 3-97 < 온라인 스케줄 수신 > 클래스 다이어그램

(아) <운행실적관리> 클래스 다이어그램

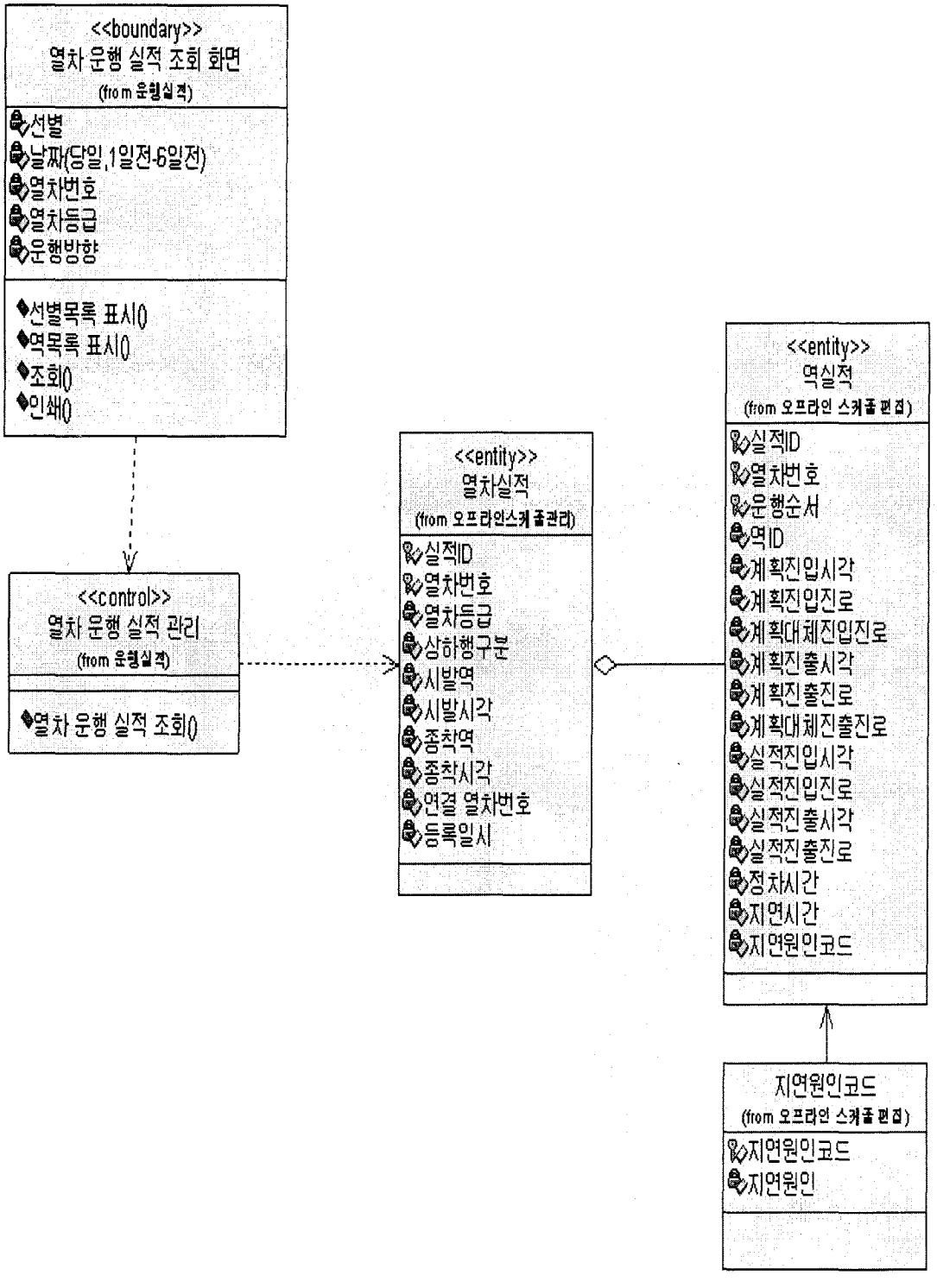


그림 3-98 운행실적관리 클래스 다이어그램

(자) <열차스케줄 정보관리> 클래스 다이어그램

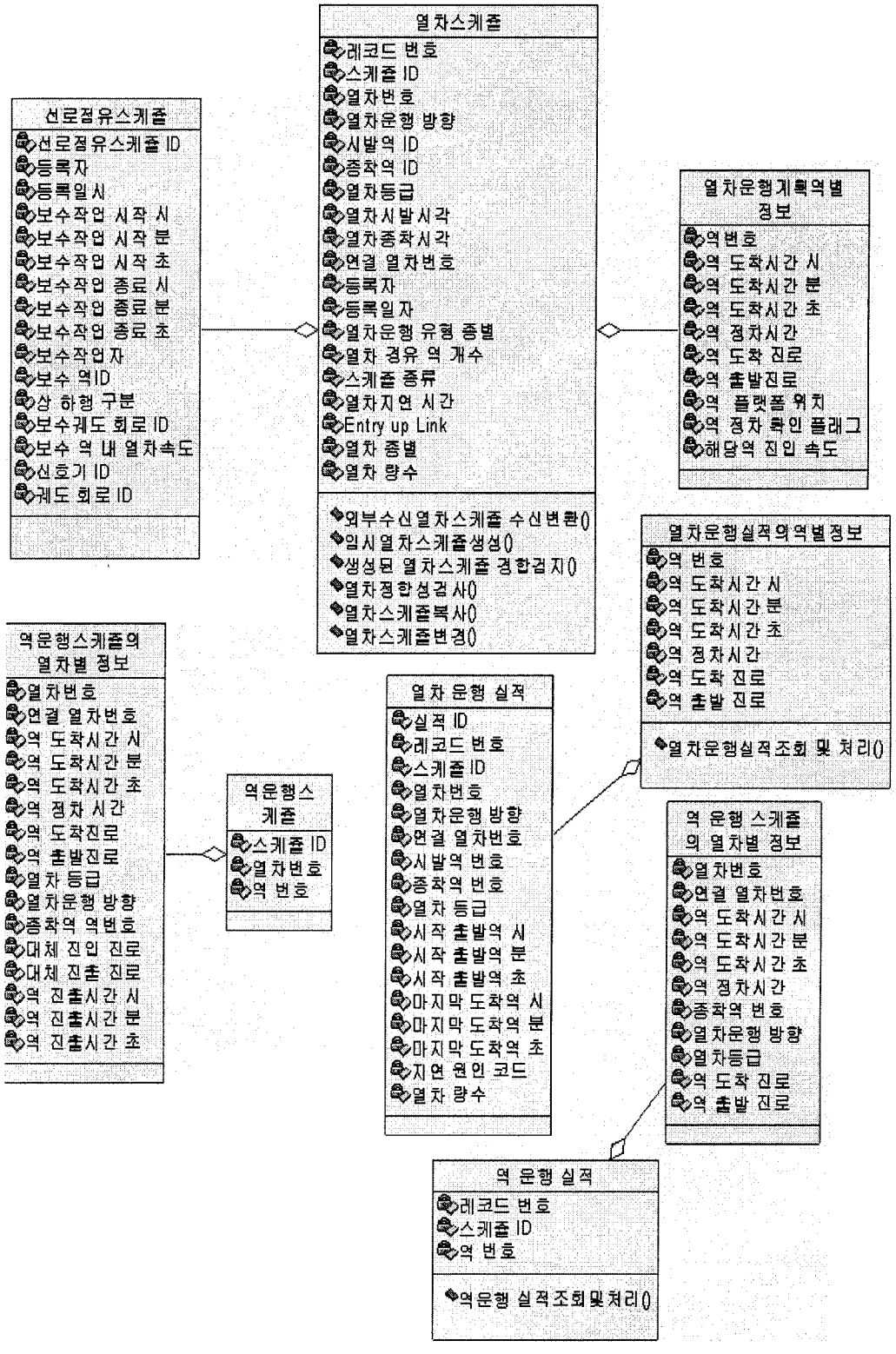


그림 3-99 열차스케줄 정보관리 클래스 다이어그램

여 백

4장 외부테스터

1절 외부테스터 상세설계

1. 외부테스터 개요

외부테스터(EXT, External Tester)는 자율분산시스템 개념을 도입하여 개발한 대형·고밀도운행 철도역 자동운행제어시스템을 유지보수관리자가 테스트 모니터링과 온라인 기능 테스트를 수행할 때 필요한 시스템이다.

자율분산 역제어시스템은 대규모 지역의 각 역들에 구축될 것이다. 대규모 지역에 설치되는 시스템 구성이 많아질수록 고장발생 빈도가 증가된다. 시스템의 고장발생 대처와 시스템 유지보수 이후의 시스템 재기동에 따른 시스템 테스트 절차는 시스템의 중지 없이 이루어져야 한다. 이러한 요구조건을 만족하기 위해서는 개발되어지는 대형·고밀도운행 철도역 자동운행제어시스템 자체의 테스트 지원 기능뿐만 아니라 독립된 서브시스템의 구조를 가지는 외부테스터가 개발되어진 시스템의 신뢰성을 높일 수 있다.

외부테스터가 어떠한 서브시스템을 테스트하기 위해서는 개발된 서브시스템에 외부테스터에서 테스트를 수행할 때 지원할 수 있는 기능들이 포함되어야 한다. 이 기능들은 서브시스템의 자율분산관리자에 포함되어 있다. 테스트를 지원하기 위한 서브시스템 자율분산관리자의 주요 기능들은 다음과 같다.

- o 서브시스템을 온라인모드와 테스트 모드로 설정하는 기능
- o 서브시스템 프로세스에서 테스트 데이터를 생성하는 기능
- o 테스트모드일 경우 하부장치들(devices)에게 명령이 전달되지 않도록 하는 기능
- o 인접서브시스템의 결함을 검지하는 기능

2. 외부테스터 요구사항 도출

외부테스터 개발을 위해 시스템 요구사항을 분석하였다. 요구사항 도출을 위해 시스템의 기능적 요구사항과 비 기능적인 요구사항 등을 정의하였다.

표 4-1 외부테스터 요구사항

No	요구사항 내용	요구사항 내용 설명
1	외부테스터는 자율분산관리자와 프로세스로 구성되어진 시스템이어야 한다.	외부테스터는 데이터필드를 사용하여 데이터를 송수신하는 서브시스템의 구조와 같이 자율분산 관리자+프로세스 구성을 가져야 한다.
2	네트워크에 연결된 서브시스템간의 통신이 가능하여야 한다.	외부테스터는 데이터필드를 통해 서브시스템과 실시간 데이터 송수신이 가능하여야 한다.
3	네트워크 연결 상태 확인이 가능하여야 한다.	외부테스터는 테스트를 위해 연결된 서브시스템이 alive상태인지 확인할 수 있어야 한다.
4	휴대성이 편리하여야 한다.	외부테스터는 역과 센터에서 사용할 수 있도록 휴대성이 있는 시스템으로 구성되어야 한다.
5	시험내용을 저장할 수 있어야 한다.	외부테스터는 테스트 데이터를 저장함으로써 분석내용을 제공하여야 한다.
6	테스트 진행상황 이상 유무를 사용자가 판단할 수 있어야 한다.	테스트 진행상황을 모니터링 하여 확인할 수 있도록 하여야 한다.
7	테스트 상황이나 분석 결과물을 출력할 수 있어야 한다.	테스트 모니터링 내용과 분석내용을 출력 지원하여야 한다.
8	시험대상 서브시스템의 모드를 확인할 수 있어야 한다.	시스템의 운영모드(테스트모드, 온라인모드)인지를 확인할 수 있어야 한다.
9	온라인/테스트 데이터필드로부터 정보를 송수신해야 한다.	온라인과 테스트 모드인 각각의 서브시스템으로부터 송출된 데이터를 수신할 수 있어야 한다.
10	서브시스템의 출력 값을 비교할 수 있어야 한다.	테스트되는 서브시스템의 출력 값을 예상되는 출력 값과 비교할 수 있어야 한다.
11	비교데이터 분석 내용을 모니터에 출력하여야 한다.	분석내용을 사용자가 판단할 수 있도록 화면에 표시한다.
12	시스템 제어명령 테스트를 할 수 있어야 한다.	RCS나 TMS에서 생성되는 제어명령의 수행여부를 테스트 할 수 있어야 한다.
13	데이터 전송에 따른 응답데이터 에러를 체크할 수 있어야 한다.	요청/응답 데이터를 분석할 수 있도록 할 수 있어야 한다.
14	멀티캐스트 데이터 누락여부를 테스트 할 수 있어야 한다.	멀티캐스트 통신에 따른 데이터 누락여부를 테스트 하여야 한다.
15	새로운 버전과 구버전 소프트웨어 데이터의 일관성 테스트를 할 수 있어야 한다.	기능 확장된 데이터와 이전 버전의 기능과 신기능을 올바로 수행 하는지 테스트 하여야 한다.

3. 외부테스터의 핵심 기능

외부테스터는 대형·고밀도운행 철도역 자동운행제어시스템의 각각의 서브시스템을 테스트하기 위해 시스템 요구사항 도출을 통해서 다음과 같은 외부테스터 주요 기능을 정의하였다.

- ㅇ 출력 제한시간 검사 기능
- ㅇ Version-up 검사 기능
- ㅇ 스케줄 송수신 검사 기능
- ㅇ 경로제어 테스트 기능

가. 출력 제한시간 검사

이 테스트는 실시간 열차위치추적 서브시스템이 열차추적데이터와 역상태정보 등을 설정된 제한시간 내에 주기적으로 출력하는지를 검사한다. 만일 열차위치추적 서브시스템에서 출력된 열차위치정보와 역상태정보를 경로제어 서브시스템이 설정된 시간 내에 수신하지 못하면 자동모드상황에서는 열차경로제어 기능을 수행할 수 없기 때문에 출력 제한시간 기능검사가 필요하다.

(1) 출력 제한시간 검사의 흐름

(가) 연결대상 설정

테스트 사용자가 테스트 대상 시스템을 외부테스터 메뉴에서 'DF연결'을 선택하면 선택되어진 서브시스템을 외부테스터에 표시하여 준다. 표시된 서브시스템으로부터 데이터를 수신한다.

(나) 연결대상 시스템 모드 판단

외부테스터에서 테스트 대상 시스템으로부터 수신된 데이터를 분석하여 시스템 모드가 온라인 모드인지 테스트 모드인지를 판단하여 표시하여 준다.

(다) 기준시간 입력

외부테스터에 테스트 대상 시스템의 데이터 출력 한계시각을 정의하기 위해 테스트 사용자가 기준 시간을 입력한다.

(라) 검사 시작

테스트 사용자가 검사 시작 버튼을 클릭하면 수신되는 테스트 시스템의 출력데이터를 수신하여 화면에 표시하여 준다.

(마) 오류 검지

수신되는 데이터가 테스트 사용자가 설정한 기준시간을 초과한 경우 이를 사용자에게 확인하도록 한다.

나. Version-up 검사

(1) 연결대상 설정

테스트 사용자가 독립된 테스트 대상 구버전 시스템과 신 버전 시스템을 데이터필드에 연결한다. 이때 구버전 시스템을 온라인모드로 실시간으로 운영되는 시스템을 연결하고 새롭게 버전업되어진 신 버전 시스템을 테스트 모드로 설정하여 테스트 한다.

(2) 버전 정보 표시

외부테스터에 테스트 대상으로 연결된 구버전 시스템과 신 버전 시스템의 버전 정보를 화면에 표시한다.

(3) 양쪽 출력 값 비교

구버전 온라인 데이터와 신 버전 테스트 출력 값이 동일한지 검사할 수 있도록 외부테스터 화면에 표시한다.

(4) 테스트 분석

신 버전의 출력 값이 구버전과 동일하면 에러가 없는 것으로 판단하며 동일하지 않는 출력 값은 그 내용을 화면에 표시하여 에러를 분석한다.

다. 스케줄 송수신 검사

(1) 테스트 스케줄 데이터 수신

센터시스템에서 전송된 데이터를 역에서 수신 받으며 이를 테스트하기 위해 외부

테스터에서도 그 데이터를 받는다.

(2) 역 응답 데이터 수신

역시스템은 센터 스케줄을 받고 그 응답데이터를 센터로 전송하며, 이 데이터를 외부테스터에서도 받는다.

(3) 두 가지 데이터 비교

센터에서 송신한 데이터와 역에서 응답한 데이터간의 누락여부 등을 비교한다.

라. 경로제어 검사

(1) 연결대상 설정

테스트 대상 시스템과 연결하는 내용코드를 등록하고 해당 데이터필드에 연결한다.

(2) 검사시작

자동진로일 경우 외부테스터는 경로제어 검사의 이상 유무를 판단하기 위해 RTPS에서 열차추적데이터를 수신 받는다. 또한 테스트 모드인 RCS의 자동진로 제어 상태를 수신대비 출력상태를 테스트 한다.

수동진로일 경우 외부테스터는 TMS에서 출력되는 제어정보를 RTPS에서 수신하고 수신된 정보에 따라 RTPS가 하위 장치(LDTS)쪽으로 정보를 전송할 때 함께 EXT쪽으로 정보를 전송한다.

(3) 경로제어 데이터 분석

자동진로일 경우 열차추적 대비 경로제어 명령이 시간적인 누락 없이 출력되었는지를 분석한다.

수동진로일 경우 사용자의 명령이 올바르게 출력되었는지 분석한다.

4. 사용자 인터페이스

가. GUI 전체 인터페이스의 구조

GUI 프로세스가 실행되면 그림 4-1과 같은 화면을 제공한다.

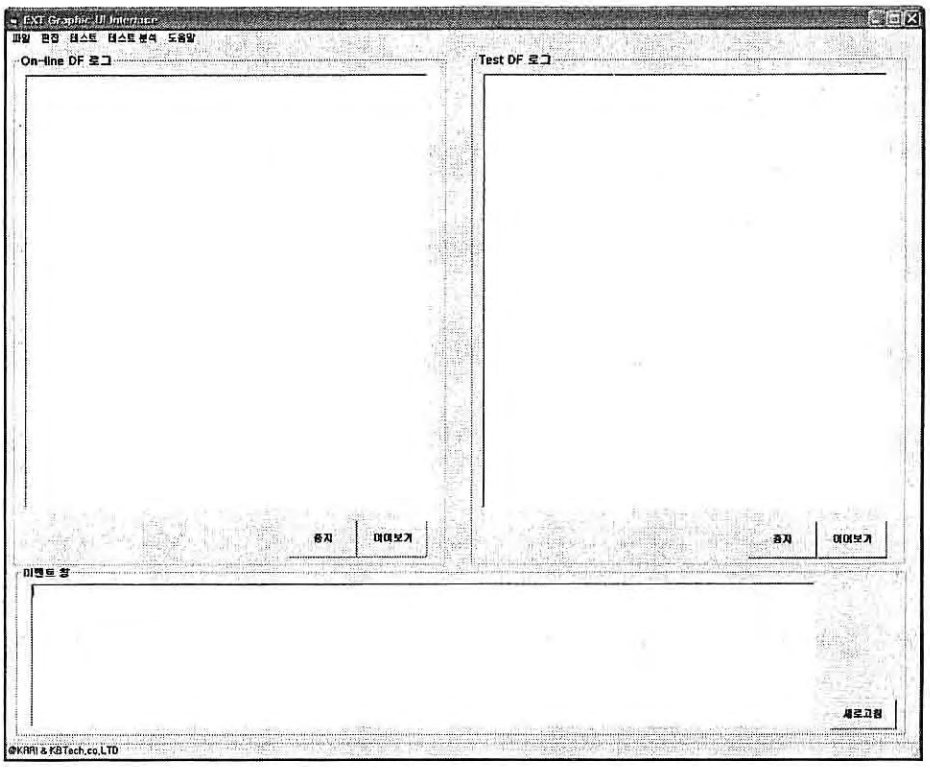


그림 4-1 외부테스터 초기화면

(1) 메뉴바

사용자 GUI 화면의 모든 기능을 파일, 편집, 테스트, 테스트 분석, 도움말을 통해서 사용한다.

(2) 데이트 데이터필드 로그 창

시스템은 테스트 데이터필드에서 수신되는 테스트 데이터 로그를 테스트 데이터필드 로그 창을 통해 출력한다. 중지버튼을 클릭하면 출력되는 내용을 잠시 멈춘다. 이어보기는 출력이 멈춘 내용을 다시 보여준다.

(3) 온라인 데이터필드 로그 창

시스템은 온라인 데이터필드에서 수신되는 온라인 데이터 로그를 온라인 데이터필드 창을 통해 출력한다.

(4) 이벤트 창

이벤트 창에는 사용자 지정한 테스트 항목 등을 표시한다. 새로 고침 버튼은 이벤트 창의 내용을 클리어 한다.

나. 메시지 분석

외부테스트에 수신된 데이터는 지속적으로 수신되기 때문에 수신된 데이터가 어떤 것인지 분석하기 위해 메시지 분석 창을 설계하였다.

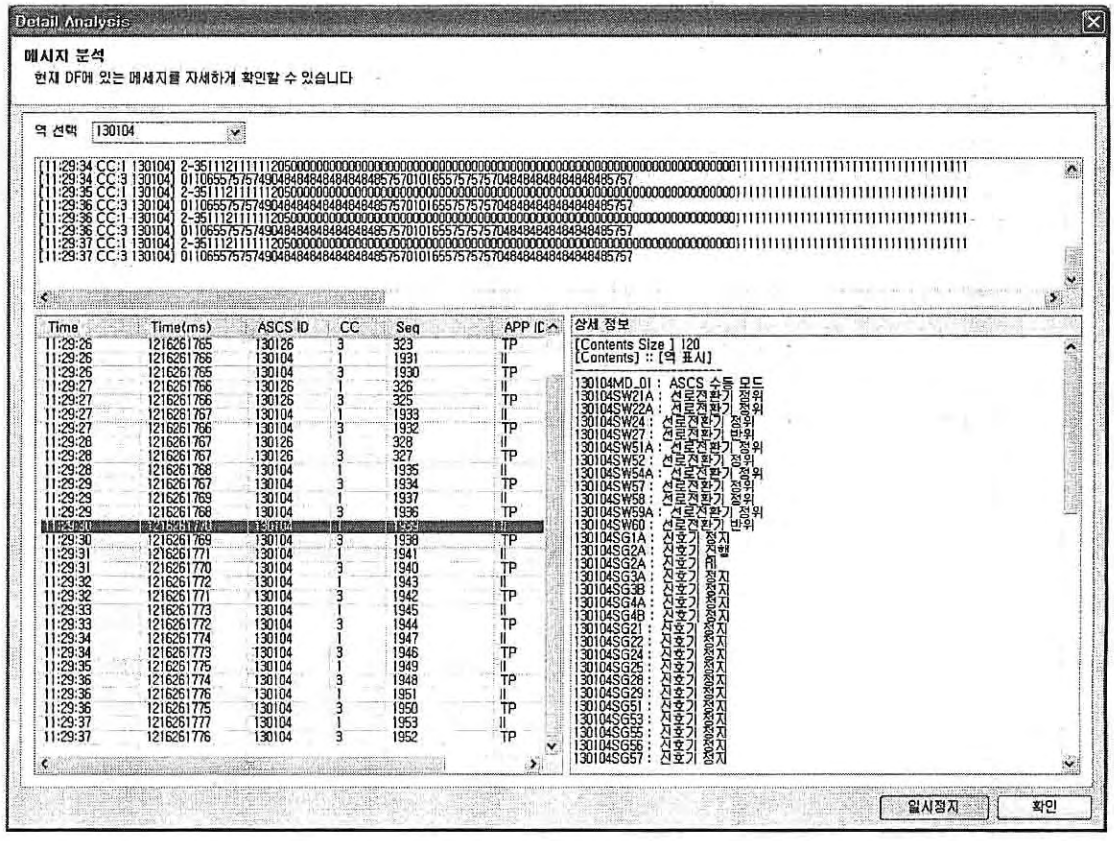


그림 4-2 메시지 분석 창

(1) 역선택

테스트 하려고 하는 역을 선택한다. 역선택 하단에 텍스트 창에 수신되는 데이터를 표시한다.

(2) 수신된 데이터 정렬 창

수신된 데이터를 수신 시간에 맞추어 정렬하고 해당 CC가 어떤 것인지 표시하여 준다.

(3) 상세정보 창

사용자가 수신된 데이터를 마우스로 클릭하면 수신된 CC의 상세 정보를 상세정

보 창에 표시하여 준다. 위의 그림4-2에서는 선택된 CC=1이 역표시정보라는 것을 알 수 있으며 역표시정보의 상태 구성을 알 수 있다.

다. 파일 메뉴

(1) 데이터필드 연결

데이터필드 연결화면은 메뉴에서 테스트 내용인 출력 제한시간 검사, Version up 검사, 스케줄 송수신 검사, 경로제어 검사 이외의 특정 서브시스템 로그 파일을 수신하여 모니터링 할 경우 사용한다.

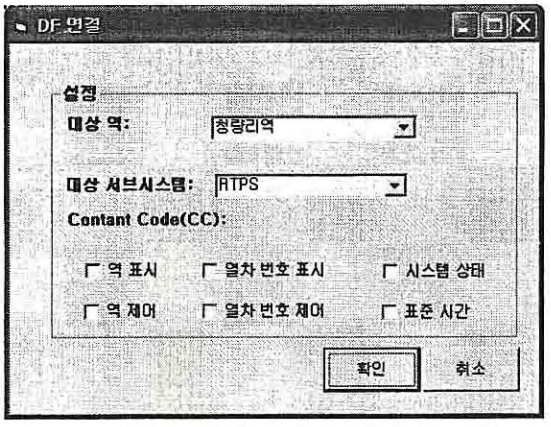


그림 4-3 데이터필드 연결 화면

(2) 출력 제한시간 구성

(가) 테스트 실행

메뉴에서 '출력 제한시간' 검사를 선택하면 그림 4-4와 같이 실행된다.

(나) 시스템 환경

'출력 제한시간' 검사 창이 실행되면 '테스트 설정'박스에서 연결 대상역과 연결 대상 서브시스템을 선택한다. 사용자가 선택할 수 있는 대상 시스템은 RTPS와 RCS이다.

'시스템모드'는 테스트 되는 시스템이 테스트 모드인지, 온라인 모드인지를 대상 서브시스템 측면에 표시하여 준다.

환경설정 창에 '기준시간설정'은 EXT시스템을 구동시키지 전에 테스트 되는 시

스템 데이터 출력 시간 한계를 사용자가 정의할 수 있다. 만일 사용자가 기준시간 설정에 값이 입력하지 않으면 '오류검지' 및 오류 확인 버튼은 활성화 되지 않는다.

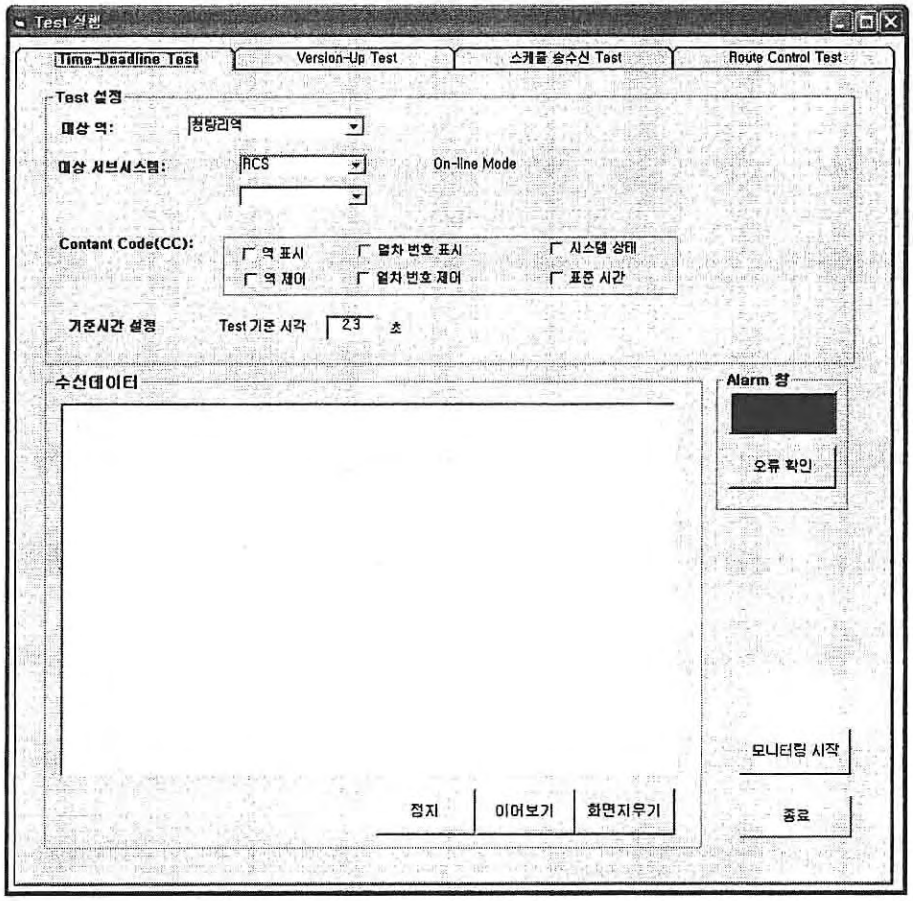


그림 4-4 출력 제한시간 검사 화면

(다) 모니터링 시작과 종료

기준시간 설정 후 '모니터링 시작' 버튼을 클릭하면 '수신데이터' 창으로 데이터 수신을 시작한다. '종료' 버튼은 출력 제한시간의 검사를 종료하고 화면이 닫힌다.

(라) 알람창

'오류검지'에는 검사가 시작되어 수신되는 데이터가 사용자가 설정한 기준시간 내에 수신되지 못할 경우 '오류검지' 문자가 반전되어 사용자에게 이상상태를 표시하여 준다.

'오류확인'은 오류검지가 발생한 경우 오류 데이터의 내용을 분석하고 싶을 경우

클릭한다. 클릭하면 시간 분석 화면 창이 출력된다.

(마) 수신데이터 창

테스트 되는 시스템에서 전송된 데이터를 수신데이터 창에 표시한다.

(바) 정지, 이어보기, 화면지우기 버튼

EXT가 데이터를 수신 중에 사용자가 정지하여 데이터를 볼 수 있고 수신되는 데이터를 계속 모니터링 하고 싶을 경우 이어보기 버튼을 클릭하며 수신데이터 창에 수신된 데이터를 지우고 싶을 경우 화면 지우기 버튼을 클릭한다.

(사) 시간 분석 화면

RTPS의 출력시간 검사일 경우 'RTPS 출력 로그 데이터' 창에만 출력된다. RCS의 출력시간 검사일 경우 RTPS에서 출력된 데이터와 이를 받아 제어명령을 내린 시간 데이터의 쌍을 RTPS출력 로드 데이터창과 RCS출력 로그 데이터창에 출력한다. '데이터 분석' 버튼을 클릭하면 그림 4-5와 같이 출력 제한시간 검사 화면이 표시된다.

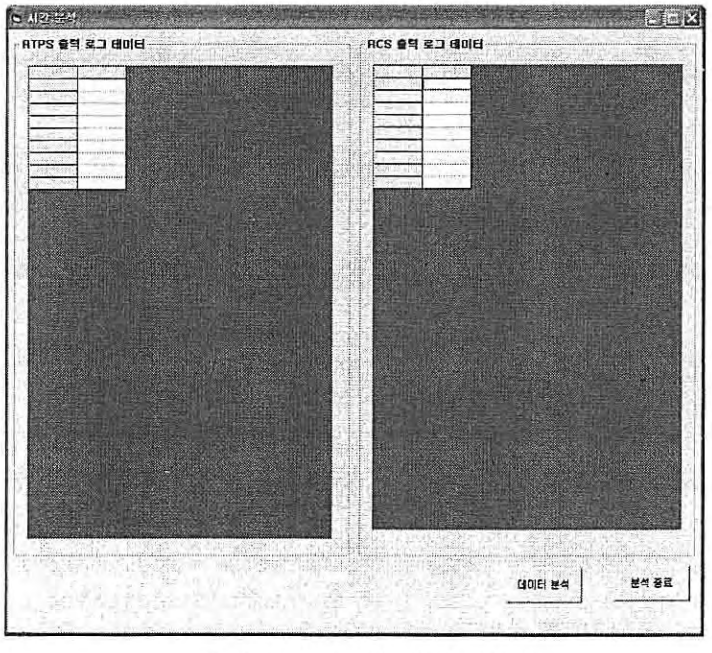


그림 4-5 시간 분석 화면

(3) Version 검사

(가) 테스트 설정

테스트를 위해 대상역과 대상서브시스템을 선택할 수 있다. 또한 컨텐츠 코드에서 수신 받고자 하는 데이터 종류를 선택할 수 있다.

(나) Version up 검사 화면

서브시스템 버전정보에서는 대상서브시스템이 어떤 서브시스템인지 표시하며 또한 구버전과 신 버전의 버전 정보를 표시한다.

(다) 모니터링 시작, 비교분석, 종료버튼

모니터링 시작 버튼이 클릭되면 두 창에 각 서브시스템으로부터 수신된 데이터를 창에 표시하여 준다.

'비교분석' 버튼을 클릭하면 그림 4-7과 같이 비교분석 화면을 표시하게 된다. '종료'버튼이 클릭되면 테스트 창이 종료된다.

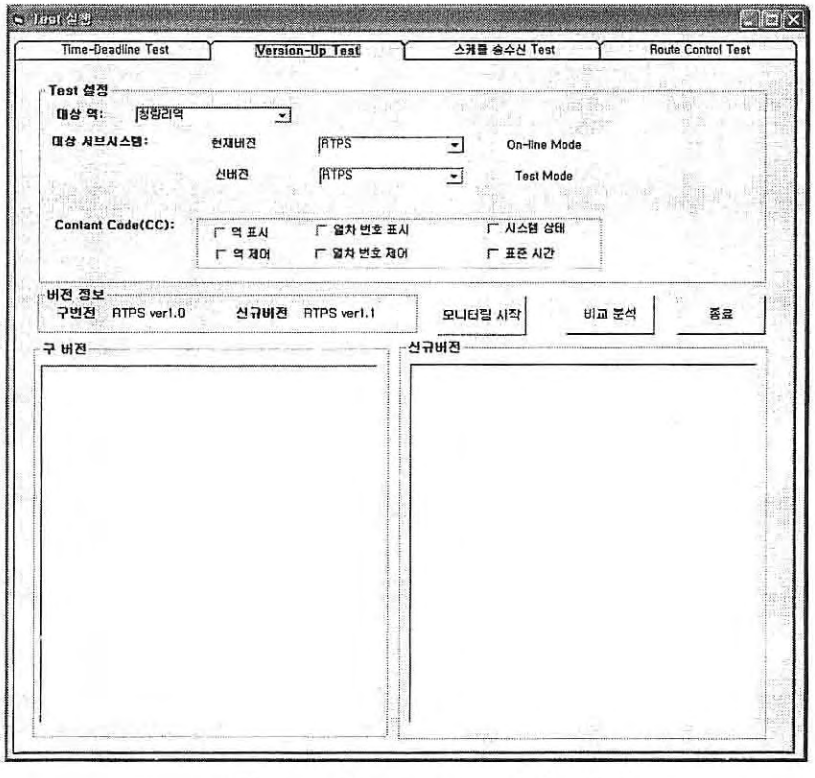


그림 4-6 Version up 검사 화면

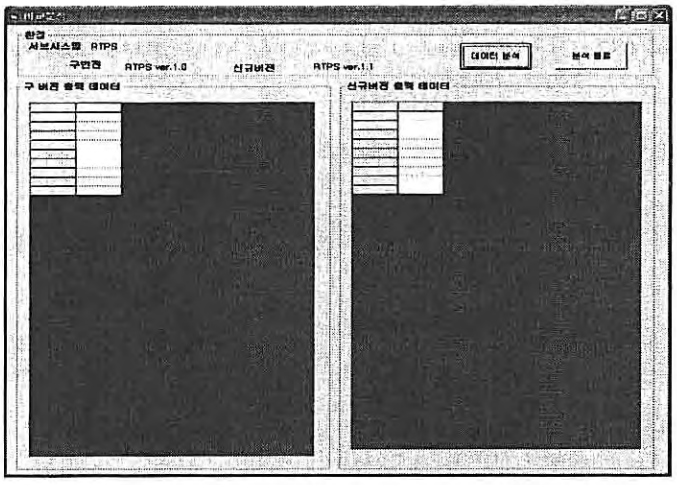


그림 4-7 버전업 검사 비교분석 화면

(4) 스케줄 송수신 검사

(가) 스케줄 송수신 검사 화면

스케줄대상 시스템에는 수신대상 시스템을 표시한다. 즉, 스케줄에 수신하므로 역에 있는 시스템에서 수신하는 시스템을 표시한다. 송신대상시스템 센터에서 스케줄을 관리하는 시스템이다.

'모니터링 시작' 버튼을 클릭하면 스케줄 송신메시지(센터->역)의 화면이 표시된다. 스케줄 수신정보에는 각 역의 서브시스템으로부터 응답이 되는 데이터가 표시된다.

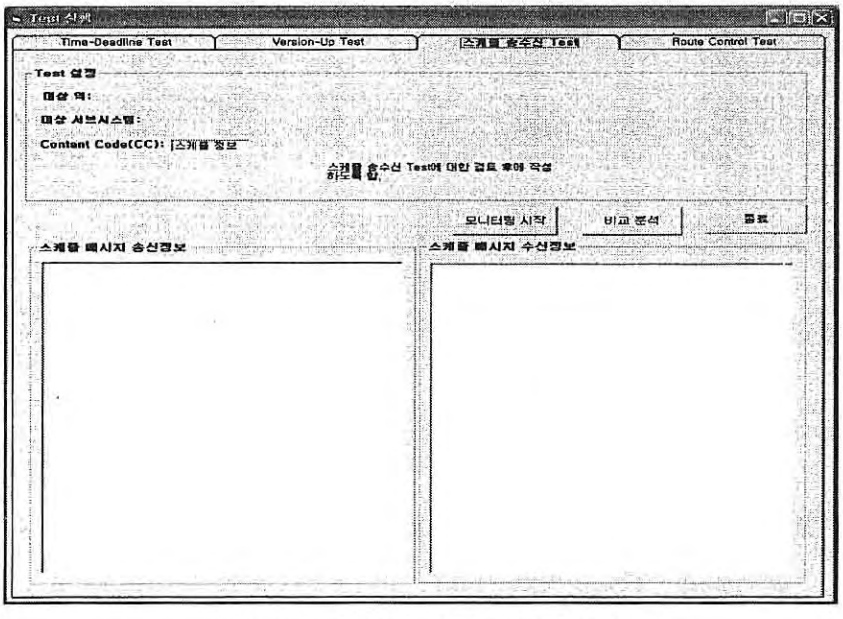


그림 4-8 스케줄 송수신 검사 화면

(나) 데이터 누락 검사 화면

그림 4-8의 비교분석 버튼을 클릭하면 그림 4-9의 데이터 누락 검사 화면이 표시된다. 데이터 누락 검사 화면에는 송신데이터에 응답한 수신데이터별 항목이 표시된다. 즉 한 개의 스케줄 수신에 응답한 서브시스템의 쌍이 표시된다. 누락여부는 송신에 대한 수신이 없는 항목을 반전으로 표시한다.

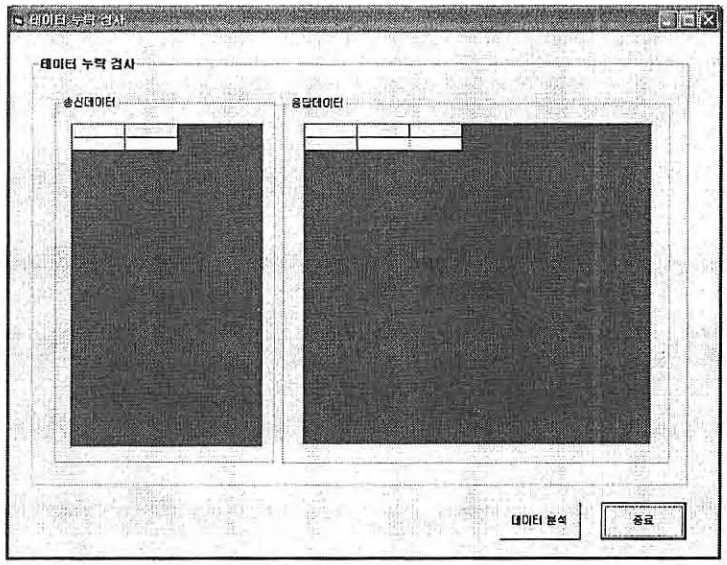


그림 4-9 데이터 누락 검사 화면

(5) 경로제어 검사

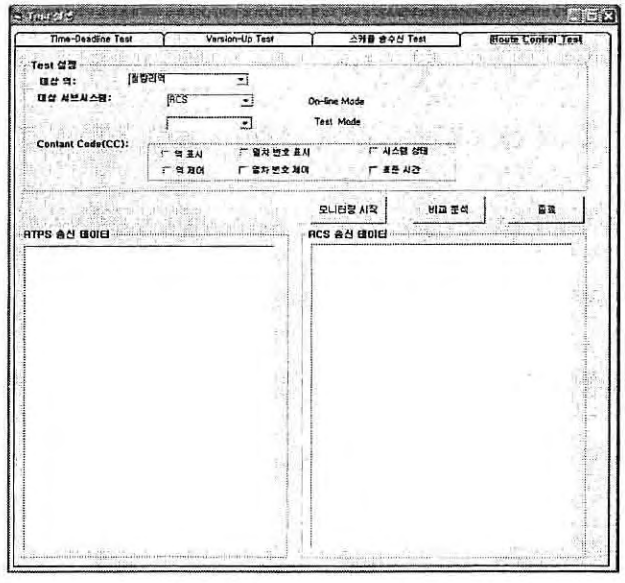


그림 4-10 경로제어 검사 화면

(가) 대상 서브시스템 선택 화면

사용자가 경로제어서브시스템에서 자동과 수동 경로제어 검사를 실시할 수 있다. 즉 사용자가 경로제어서브시스템의 '자동(RCS)'과 '수동(TMS)'를 선택한다. 만일 수동 검사일 경우 RTPS에서 수신된 열차위치정보 및 역 상태정보에 즉시 경로제어 명령을 내리는 것이 않으므로 테스트 분석의 '비교' 버튼은 반전되어 클릭할 수 없게 된다.

'검사시작'은 화면에 송수신 정보 표시를 시작한다. '검사종료'는 화면을 종료한다.

(나) RTPS 송출정보와 RCS 송출정보

'RTPS 송출정보'는 자동경로제어에 의한 경로제어 검사를 할 때 RTPS에서 수신되는 정보를 표시하여 준다. 'RCS 송출정보'는 자동에 의해 경로제어 명령이 송출된 사항을 표시하여 준다.

(다) 데이터 분석 화면

그림 4-10의 '비교 분석' 버튼을 클릭하면 그림 4-11의 데이터 분석화면이 표시된다. 즉, RTPS 송출정보에 따라 RCS가 출력한 제어명령을 하나의 쌍으로서 비교한다. 그림 4-11의 데이터분석 화면을 클릭하면 RTPS 송출에 따른 RCS 제어명령의 쌍에 이상이 생긴 데이터에 반전표시를 하여 이상 유무를 판단할 수 있도록 한다.

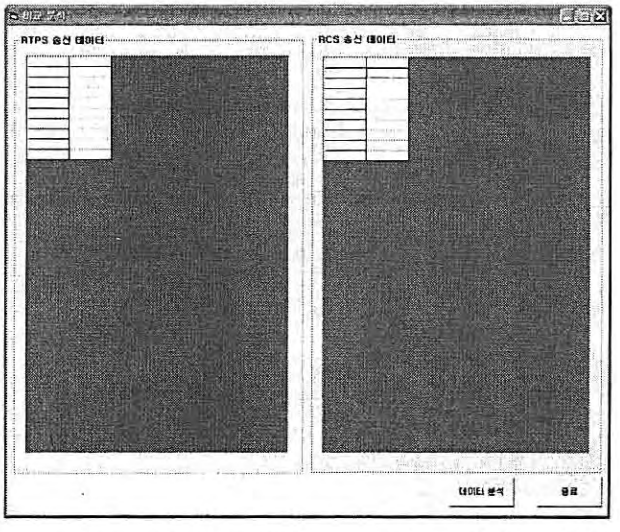


그림 4-11 경로제어 검사 비교 분석 화면

2절 종합연계시험센터에서의 기능 시험

1. 종합연계시험센터 환경

외부테스터 시험을 위하여 개발된 4개의 자율분산 역제어시스템 장비들과 자율분산 노선관리 서브시스템, 현장의 열차 이동을 모사하는 다중열차운행시뮬레이터와 신호상태를 모사하는 종합연계시험센터의 데이터필드에 연결하여 외부테스터의 핵심 기능들을 테스트 하였다.

종합연계시험센터 구성도는 그림 4-12와 같다. 라인센터의 역할을 하는 자율분산 노선관리시스템은 라인별 열차 주행상태를 모니터링 하는 라인모니터링 시스템과 라인단위의 열차 경합을 해소하는 라인경합관리 서브시스템, 라인 스케줄을 관리하는 라인정보 서브시스템으로 구성되어 있다. 자율분산 역제어시스템(ASCS)은 열차 위치를 파악하는 실시간 열차추적 서브시스템, 열차경로를 제어하는 경로제어 서브시스템과 수동제어와 화면표시 기능을 수행하는 열차모니터링 서브시스템으로 구성되어져 있다. 역과 열차의 모의환경 구현을 위해 다중열차운행시뮬레이터와 역에뮬레이터도 설치되었다.

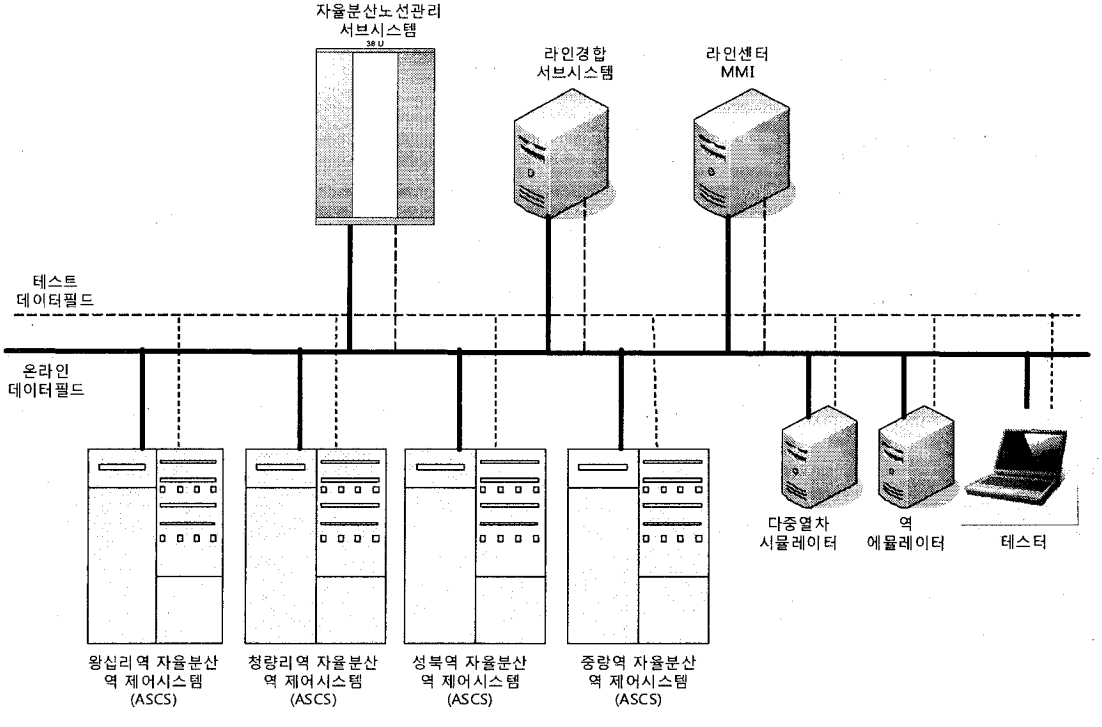


그림 4-12 종합연계시험센터 시스템 구성도

종합연계시험센터에서는 왕십리역, 청량리역, 성북역, 중랑역을 대상으로 하여 시스템을 구성하였다. 그림 4-13은 종합연계시험센터 시험실의 현장 사진이다.



그림 4-13 종합연계시험센터 시험실

2. 출력 제한시간 시험

가. 시험 개요

본 시험의 개요는 실시간 열차위치추적 서브시스템에서 주기적으로 송신되는 역 상태정보와 열차위치 정보를 열차위치추적 서브시스템에서 정해진 시간 내에 관련 데이터를 출력하는지 테스트 한다. 또한 출력 제한시간 시험은 경로제어 서브시스템에서 열차추적데이터를 주기적으로 받는지를 테스트 하였다. 만일 제한시간을 초과한 데이터를 수신하게 되면 경로제어 시스템에서는 열차상태 등의 데이터를 받지 못하게 되고 신호 설정을 할 수 없기 때문에 심각한 상황을 초래할 수 있기 때문이다.

그림 4-14는 왕십리 역 실시간 열차위치추적 서브시스템의 표시정보 관리 프로세스에서 처리한 역 표시정보 출력 데이터를 출력 제한시간 시험을 수행한 구성도

이다. 열차 주행 상황은 다중열차운행시뮬레이터를 통해 모사되었으며 LDTS를 통해 열차 주행은 역 에뮬레이터를 통해 역 상태정보 데이터를 표시하도록 하였다. 따라서 실시간 열차위치추적 서브시스템에서 데이터필드로 출력되는 데이터를 외부테스터에서 수신하여 테스트 하였다.

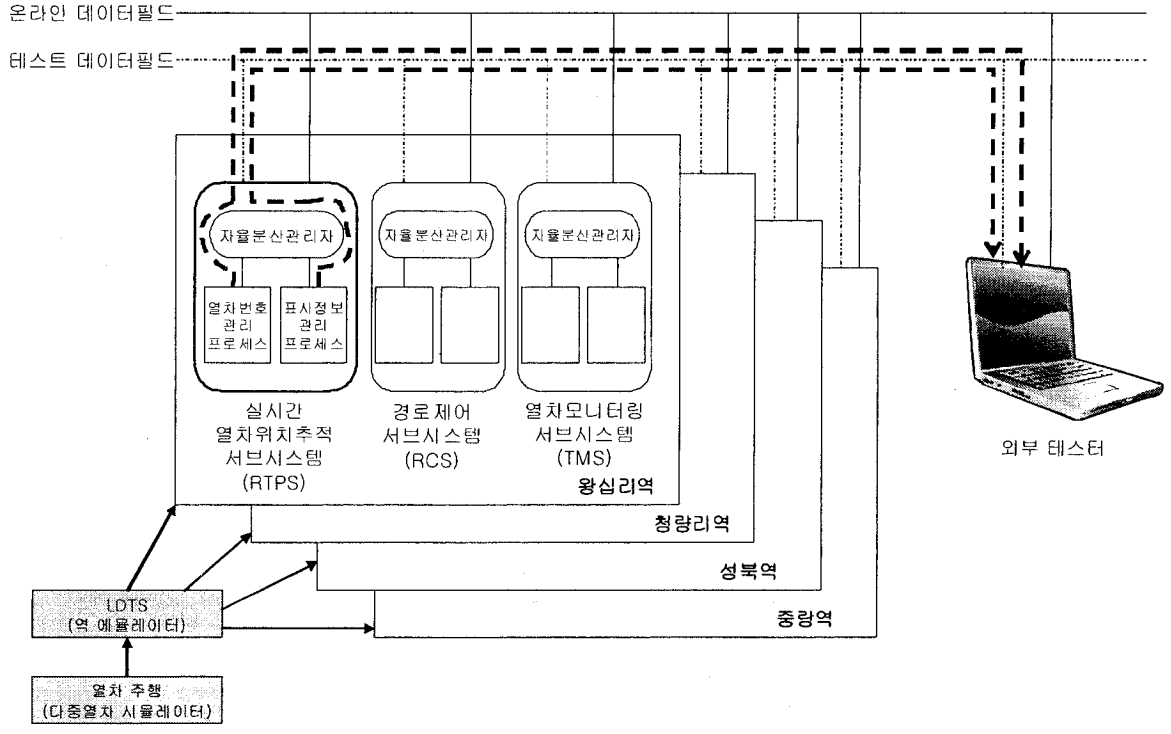


그림 4-14 출력 제한시간 시험 개요도

나. 시험 결과

출력 제한시간 시험 개요도에서와 같이 왕십리역의 역표시 정보가 주기적인 시간 내에 출력되는지를 데이터필드와 연결된 테스터를 통하여 그 정보를 수신하여 시험하였다.

그림 4-15와 같이 대상역인 왕십리역과 대상시스템인 실시간 열차위치추적 서브시스템을 선택하고 해당 내용코드(CC)를 선택하였다. 기준시간을 설정하는 텍스트 차에 테스트를 위한 기준시간을 입력하였다. 하단의 시작 버튼을 클릭하고 테스트를 수행하여 해당 데이터가 수신되는 것을 중앙의 텍스트 창에 표시되는 정보를 통해 역표시 데이터가 주기적으로 출력되는지를 확인할 수 있었다. 수신데이터는

텍스트 창에서 역번호(130126)와 내용코드 번호 1번인 역표시 데이터가 정상적으로 표출된 것을 확인하였다. 또한 수신된 데이터의 상세 정보를 확인 할 수 있도록 그림 4-16과 같이 메시지 분석 창에서 해당 내용코드를 선택하면 역 표시정보의 상세 정보인 ASCS의 상세모드가 수동모드이며 선로전환기와 신호기의 상세정보를 확인하였다.

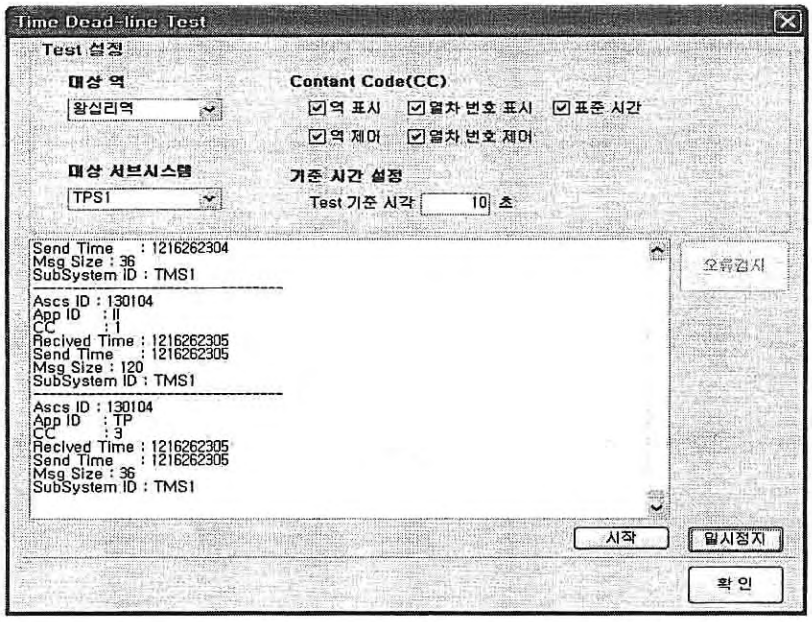


그림 4-15 주기적 통신 제한시험 화면

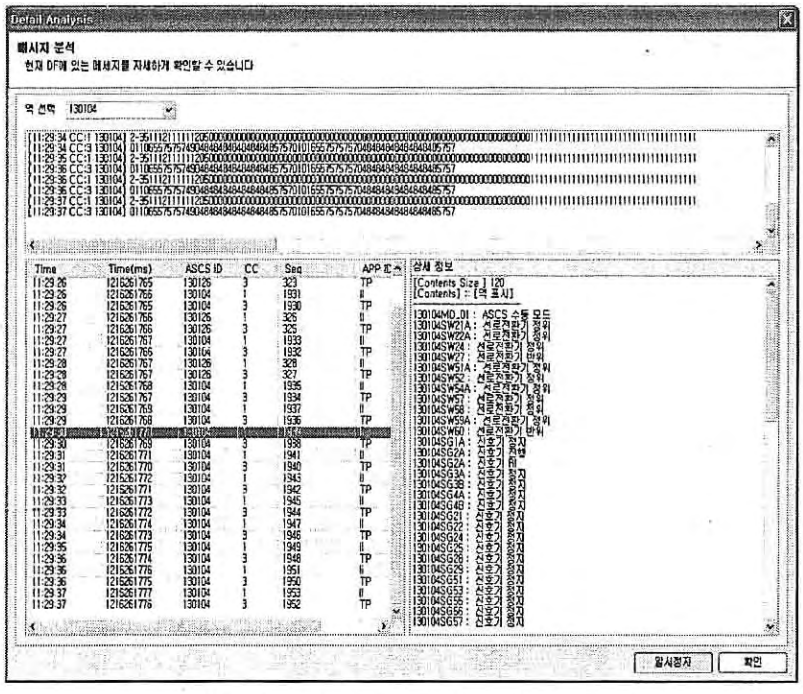


그림 4-16 역표시 데이터의 상세 내용

3. 스케줄 수신 시험

가. 시험 개요

열차스케줄은 메시지가 대용량이기 때문에 라인정보시스템에서 각 역으로 전송되는 스케줄에 대한 메시지 누락 검사가 필요하다. 또한 스케줄의 누락 검사뿐만 아니라 대용량 스케줄 수신시에 발생할 수 있는 타 시스템의 구동에 문제가 발생하지 않는지도 검사할 수 있다.

그림 4-17 스케줄 수신 시험 개요도 은 센터 역할을 하는 자율분산 노선관리 서브시스템에서 역의 수송관리 서브시스템으로 전송되는 스케줄 데이터를 외부 테스터에서도 함께 수신하여 데이터의 누락 검사를 수행하였다.

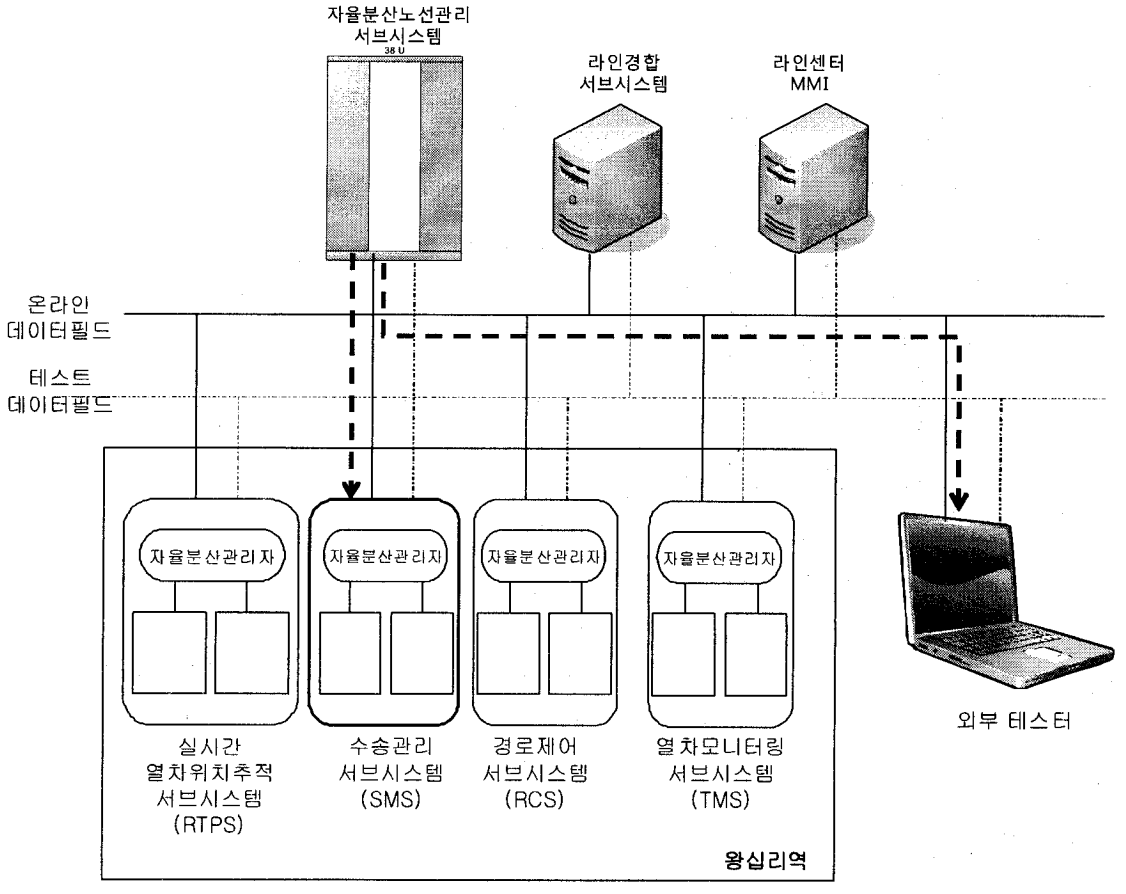


그림 4-17 스케줄 수신 시험 개요도

나. 시험 결과

역시스템에서 스케줄 데이터를 수신 받는 경우는 전체 스케줄을 수신하는 경우와 변경된 스케줄을 수신하는 경우이다. 또한 역시스템이 재가동 되었을 경우에도 센터에서 스케줄을 수신한다. 따라서 시험절차는 역시스템에 있는 모든 스케줄 데이터를 삭제하고 자율분산관리자를 재 구동하면 스케줄 데이터를 새롭게 수신하므로 수신되는 스케줄 데이터를 테스터에서 역시스템에서 수신되는 스케줄정보를 동일한 시간에 함께 수신하여 스케줄 데이터의 누락시험을 시행하였다.

그림 4-18에서 수신된 내용코드 6번인 스케줄 데이터의 상세 분석화면이다. 또한 스케줄 수신시에 타 서브시스템에 구동에 문제가 없는 것을 확인하였다.

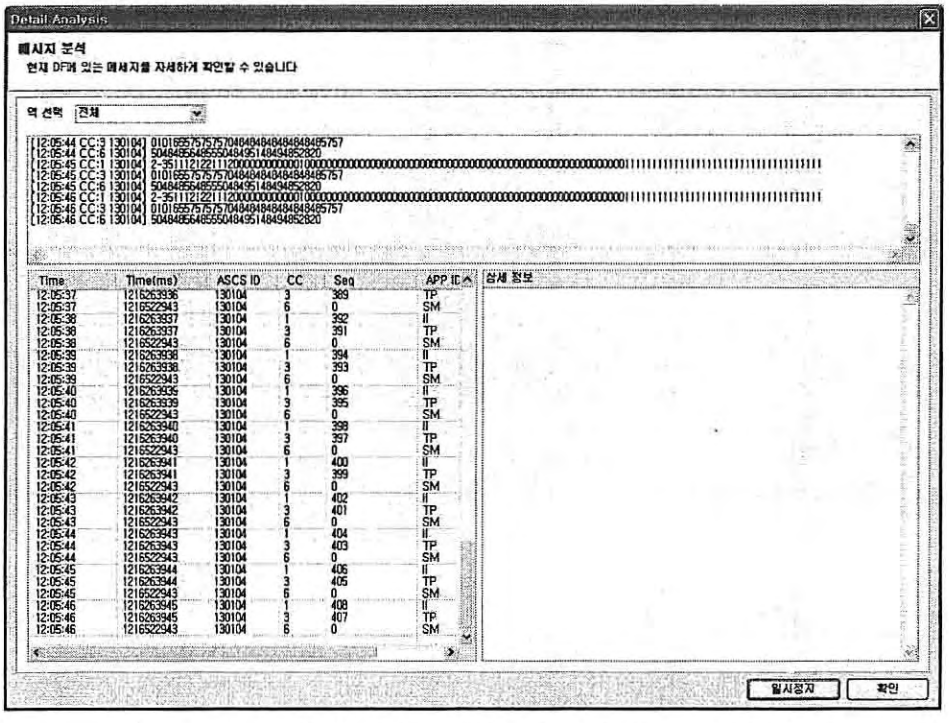


그림 4-18 수신된 스케줄 데이터의 상세 결과

4. 경로제어 시험

가. 시험 개요

경로제어 시험은 자동경로제어와 수동경로제어의 기능 이상 유무를 검사하기 위한 것이다. 자동진로제어는 경로제어 서브시스템에서 수행되고 수동진로제어는 열

차모니터링 서브시스템에서 수행된다.

그림 4-19는 수동진로 테스트 개요도이다. ASCS를 운영하는 운영자가 열차모니터링 서브시스템에서 수동진로를 설정하면 그 신호가 실시간 열차위치추적 서브시스템과 외부테스터로 테스트 데이터 필드를 통해 데이터가 전송된다. 수동 경로제어 시험에서는 외부테스터에서 수신된 경로제어 데이터를 분석하였다.

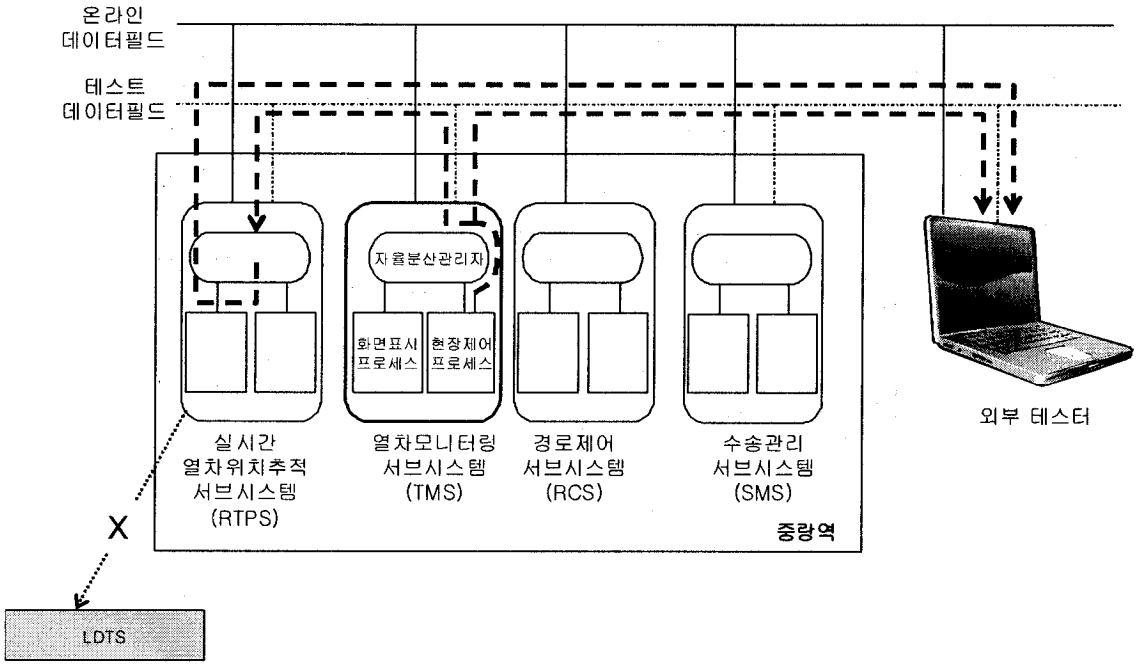


그림 4-19 경로제어 시험 개요도

나. 시험 결과

경로제어 시험은 실시간 열차위치추적 서브시스템(RTPS)에서 송출되는 역 상태정보 및 열차 위치정보를 경로제어 서브시스템에서 수신하여 경로를 제어하기 때문에 RTPS에서의 송출데이터와 이를 수신하여 경로를 설정한 경로제어 서브시스템에서 송출된 데이터를 비교하여 경로를 제어한 것을 비교하여 테스트 하였다. 그림 4-20에서와 같이 RTPS 송신데이터와 RCS송신데이터가 테스터를 통해 수신된 데이터를 비교한 결과이다. RCS 창에서 CC=2는 역 제어 명령이고 제어 명령이 수신된 것을 확인하였다. 그림 4-21은 RTPS와 RCS에서의 송출 데이터를 비교한 결과이다.

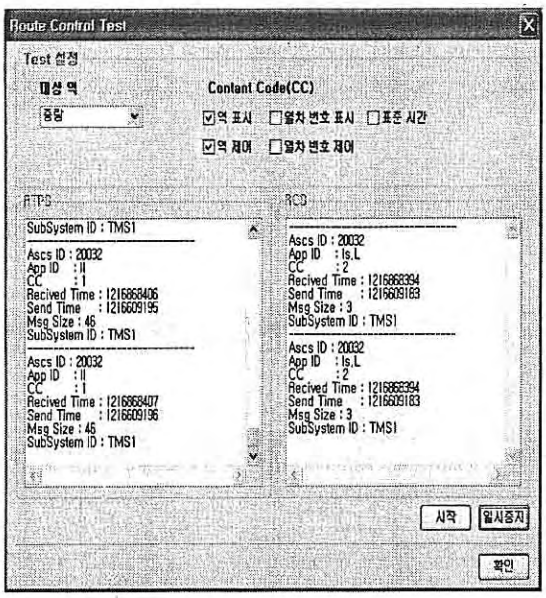


그림 4-20 수동진로 테스트 결과

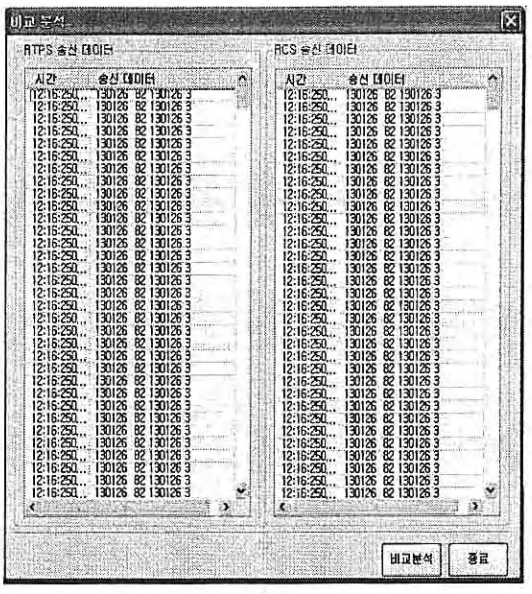


그림 4-21 경로제어 송신데이터 비교 결과

5. 버전업 시험

가. 시험 개요

자율분산 역제어시스템의 특징 중 하나가 단계적인 구축이 가능한 것이다. 이는 시스템 구축 후에 발생되는 추가적인 사용자 요구사항 등을 수집하여 시스템에 반영할 수 있다. 이러한 요구사항을 만족하기 위한 버전업 시험을 하였다.

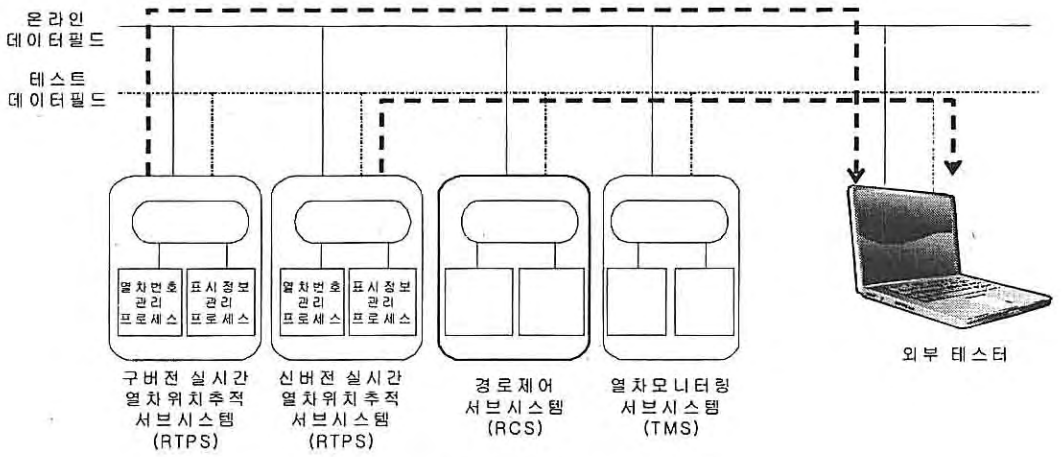


그림 4-22 버전업 시험 개요도

그림 4-22는 버전업 시험의 개요도이다. 버전업 시험은 온라인 테스트를 사용할 수 있는데 시스템을 중지하지 않고 테스트하기 위해 구동중인 온라인 시스템 실시간 열차위치추적 서브시스템(RTPS)과 테스트 모드로 설정된 신버전 실시간 열차위치추적 서브시스템(RTPS)을 온라인 데이터필드와 테스트 데이터 필드로 연결한다. 온라인 상태의 구버전 RTPS에서 출력되는 실시간 정보와 신버전 RTPS에서 출력되는 테스트 데이터 정보를 외부테스터에서 수신하여 신버전 시스템이 구버전 시스템과 동일한 기능을 수행하는지 테스트 하였다.

나. 시험 결과

자율분산 역제어시스템 기능 확장에 대한 버전업 시험결과는 그림 4-23과 같다. 대상역은 중랑역을 선택하고 대상서브시스템은 RTPS를 선택하였다. 온라인 모드인 구버전과 테스트 모드인 신버전 RTPS의 버전정보가 나타난다.[2)]

구버전 텍스트 창에 구버전에서 RTPS에서 출력된 데이터에 중랑역 번호, 어플리케이션 정보, 내용코드 1번인 역표시 정보 등이 표시되었다. 신버전 RTPS에서 출력된 데이터도 구버전 시스템에서 출력한 데이터와 일치되는 것을 알 수 있다.

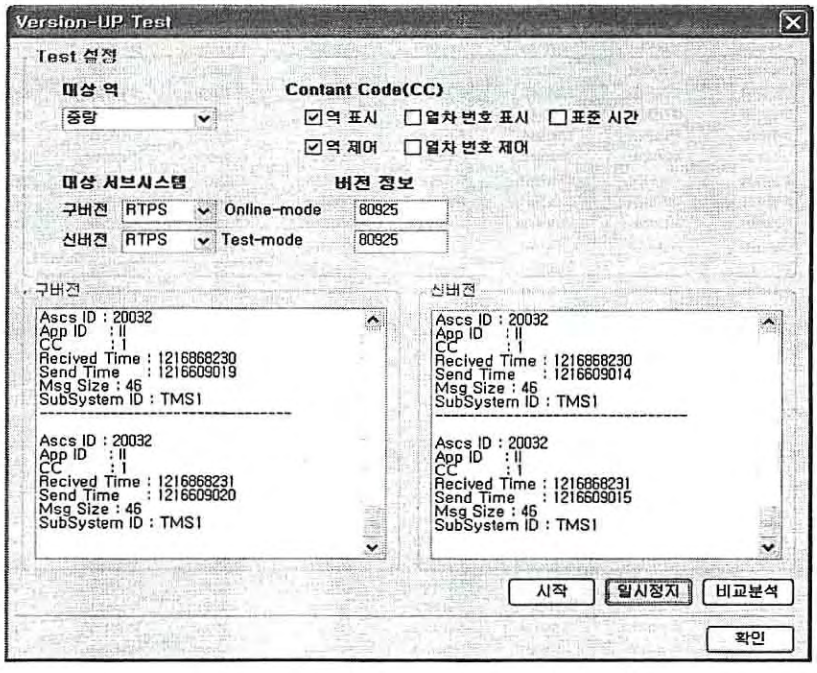


그림 4-23 버전업 테스트 시험 결과

2) 버전정보가 같은 이유는 실제 버전업된 시스템으로 테스트 하지 못하고 동일한 모듈이 설치된 독립 서브시스템을 온라인 모드와 테스트모드로 설정하여 테스트하였음.

여 백

5장 종합연계시험센터

1절 종합연계시험센터 개발

1. 종합연계 시험계획 작성

가. 테스트 개요

일반적으로 테스트의 단계적 분류 방법은 단위, 통합, 시스템, 인수, 설치 테스트로 구분되며, ADTMS의 종합연계시험센터에 대한 시험은 시스템 테스트 단계로 구분된다. 일반적인 테스트 프로세스는 그림 5-1과 같은 흐름을 가진다.

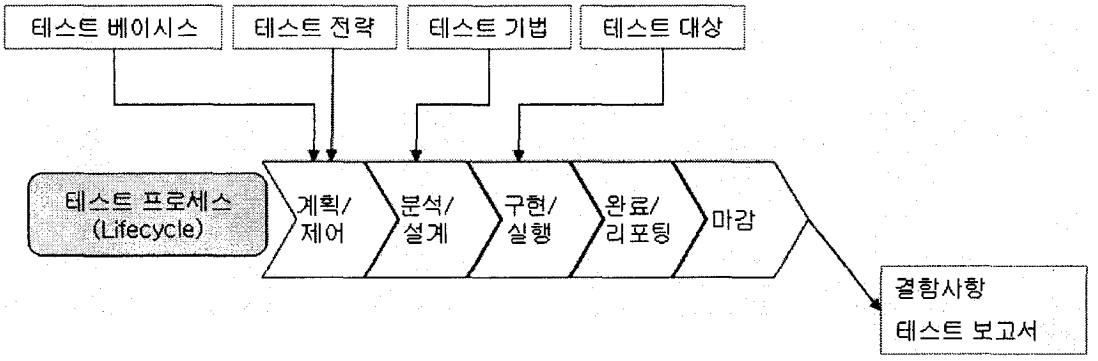


그림 5-1 테스트 프로세스 구성요소

테스트 프로세스에 따라 테스트가 진행되는 경우 이를 뒷받침할 수 있는 다양한 입력문서와 출력문서들이 산출된다.

나. 종합연계시험센터 시스템 테스트 계획서

(1) 개요

ADTMS 시스템 테스트 계획서는 다음 목적을 지원하도록 한다.

(가) 시스템 시험을 준비하고 수행하는데 필요한 활동을 상세히 설명한다.

(나) 관련된 인원에게 그들이 수행하여야 할 작업과 그 일정을 안내한다.

(다) 계획서를 준비하는데 사용하는 정보의 출처를 정의한다.

(라) 시스템 테스트를 수행하는데 필요한 시험 도구와 환경을 정의한다.

이 계획서의 주안점은 ADTMS 시스템의 사용자 테스트를 위하여 ADTMS를 구성하는 각 구성품/서브시스템의 결합을 시험하는 것이다.

시스템 테스팅의 일정은 마스터 테스트 계획서에 따른다.

계획서는ADTMS 프로젝트와 직접적으로 연관되어 있는 시스템의 시험에 적용된다. 이 테스트 계획서는 프로그램 및 작업관리 뿐만 아니라 운영자 및 사용자 절차를 포함하는 ADTMS의 시스템 테스트에 관한 모든 것을 다룬다.

(2) 테스트 항목

ADTMS를 구성하는 모든 항목은 시스템 테스트 내에서 테스트 된다.

다음의 목록이 테스트 되어야 하는 항목들이다

- 기능 테스트
- 인터페이스 테스트
- 성능 테스트 및 volume test
- 고장 및 재기동 테스트

시험 대상 항목은 다음과 같다.

(가) 프로그램 목록

테스트 대상 프로그램 목록.

(나) 작업 제어 절차

응용 프로그램 및 유틸리티 프로그램에 대한 제어 절차.

(다) 사용자 절차

ADTMS 시스템 처리 절차 참조 매뉴얼에 명시한 온라인 절차를 시험한다.

(라) 운영자 절차

ADTMS 운영자 참조 매뉴얼에 명시한 절차를 시험한다.

(3) 테스트 대상 특성

다음의 항목이 응용프로그램의 테스트에 주안점을 두어야 할 테스트 대상 기능

목록이다.

- □ 자율분산관리자 기능
- □ 실시간 열차추적(RTPS) 서브시스템 기능
- □ 경로제어(RCS) 서브시스템 기능
- □ 열차모니터링(TMS) 서브시스템 기능
- □ 스케줄관리(SMS) 서브시스템 기능
- □ 역경합관리 기능
- □ 외부테스터(EXT) 서브시스템 기능
- □ 라인정보(LIS) 서브시스템 기능
- □ 라인모니터링(LSS) 서브시스템 기능
- □ 센터경합관리(GCDRS) 서브시스템 기능
- □ 테스트되어야 되는 컴포넌트/서브시스템
- □ 테스트 대상 인터페이스

ADTMS 프로젝트의 모든 입/출력 인터페이스가 테스트되어야 한다.

- ADM
- LMS-SMS
- RTPS-RCS
- RTPS-TMS
- RTPS-LDTS(외부 인터페이스)

인터페이스 시험 중 외부 시스템과의 인터페이스를 제외한 모든 인터페이스 시험은 자율분산관리자를 통한 인터페이스이므로, 인터페이스 테스트의 주안점은 자율분산관리자의 기능에 대한 테스트가 될 것이다.

(4) 테스트 대상이 아닌 특성

(가) 테스트 제외 컴포넌트/서브시스템

역시스템 중 LDTS는 이미 현장에 설치되어 있는 시스템으로 이 시스템과의 인터페이스 시험을 시스템 테스트 단계에서 직접 수행할 수 없으므로 LDTS는 시뮬레이터를 대신 사용하여 시험한다. LDTS에 대한 시뮬레이터는 역 연동장치에 대

한 시뮬레이터와 열차의 움직임을 발생시키는 시뮬레이션을 포함한다.

(5) 접근 방법

프로그램은 모든 치명적 결함이 제거된 이후에 시스템 테스트가 시작되어야 한다. 하나의 프로그램은 프로그램의 테스트를 방해하지 않는 최대한 두개 이내의 중요 결함이 있는 것이 시스템 테스트 단계에 허용된다.(마스터 테스트 계획서 참조)

(가) 테스트 계획 및 준비

테스트 조건 목록이 포함되어 있는 시스템 테스트 사양서는 ADTMS 프로젝트를 기준으로 한다. 기능 목록과 그 기능에 해당되는 테스트 절차는 테스트 프로세스의 소개에 따른다.

(나) 테스트 선행조건

시스템 테스트 조건은 시스템 테스터에 의해 구비되어 있어야 하며, 시스템 테스트 절차서의 생성보다 앞서서 이루어진다. 시스템 테스트 조건은 사용자에 의해 선행되어 검토가 완료된 기능 목록에서 추출된다.

테스트 절차는 가능하면 순서대로 수행되어야 하며, 높은 중요성을 가지는 테스트 조건들이 먼저 테스트되어야 한다.

(다) 제약사항

- 역에서 ADTMS와 직접 SERIAL로 인터페이스 되는 LDTS와는 연결 시험이 불가능하므로 LDTS 및 이 하위의 연동장치를 에뮬레이션 하는 시스템(역에뮬레이터 및 다중열차운행시뮬레이터)이 LDTS와의 인터페이스를 대신한다.
- LMS 구동을 위한 서버(rp4440)가 준비되어 있어야 한다. 서버에는 응용 프로그램 구동을 위한 O/S, DB가 설치되어 있어야 한다.
- PC 기반의 컴퓨터에 O/S가 설치되어 있어야 한다.
- 각 서브시스템의 목적에 맞는 데이터베이스와 Configuration의 설정이 완료되어 있어야 한다.

- 각 서브시스템의 데이터베이스 보안 및 접근 제한은 반드시 정의되어 있어야 하며, 보안이 안전하다는 것이 증명되어야 한다.

다. 시험 시나리오 작성

종합연계시험센터의 시험은 개별 프로세스 시험과 각 서브시스템간의 상호 인터페이스, 그리고 각 서브시스템간의 기능 연계 시험에 대해서 시나리오를 작성 후 시나리오에 따라서 시험이 이루어진다. 시험 시나리오는 각 테스트의 종류에 따라 분류된다. 시험 시나리오는 유즈케이스가 기반이 되며, 시스템의 기능 및 성능에 영향을 미치는 여러 가지 요소들을 조합하는 형식으로 작성된다.

시험 시나리오는 한 개 이상의 테스트 케이스가 필요하며, 필요한 경우 각 테스트 케이스간의 순서에 따라 수행되어야 한다.

(1) 테스트 시나리오 작성 예(오프라인 스케줄 다운로드)

1.1. 오프라인 스케줄 다운로드

1.1.1. 테스트 절차 사양 ID

ID : ADTMS_TPS_IT005.

1.1.2. 목적

이 사양서는 LIS에서 ASCS로 역별 열차 운행계획 파일을 전송하는 기능에 대해 테스트를 위해 필요한 각 단계와 테스트 명세서에 기술된 테스트 케이스의 실행에 대하여 설명한다. 이 테스트 절차에 사용되는 테스트 케이스는 다음과 같다 : - ADTMS_TCS_IT005

1.1.3. 특별 요구사항

테스터는 Windows XP와 UNIX 시스템 사용에 익숙하여야 한다.
테스터는 자율분산관리자의 특성에 대해 알고 있어야 한다.
테스터는 Altibase 데이터베이스에 대해 알고 있어야 한다.
테스터는 MANTIS 사용법을 알고 있어야 한다.

1.1.4. 절차 단계

1.1.4.1. 로그(Log)

테스트 실행 결과, 목격된 사고, 그리고 다른 어떤 테스트에 관련된 이벤트에 대한 특정한 방법이나 로깅 형식을 설명한다.

1.1.4.2. 로그(상황 기록)

테스트 실행 결과는 IEEE829에 기술된 대로 테스트 로그에 기록한다. 로그 기록에 관

한 사항은 마스터 테스트 절차 사양서를 참조한다.

1.1.4.3. 설정

테스트 절차 사양 ID ADTMS_TPS_IT001의 테스트가 완료되어 있어야 한다.
LIS와 SMS의 환경 정보가 설정되어 있어야 한다.
LIS의 열차운행계획 전송 기능이 구현되어 있어야 한다.
ASCS의 SMS 기능이 구현되어 있어야 한다.
테스트를 시작한다.

1.1.4.4. 시작

각 서브시스템의 자율분산관리자를 실행한다.
LIS의 데이터베이스를 초기화한다.
SMS의 역 스케줄 데이터베이스를 초기화한다.
LIS의 스케줄관리 기능을 시작한다.
SMS를 시작한다.

1.1.4.5. 진행

테스트 케이스를 정해진 순서대로 실행 한다.
테스트 케이스는 의존성이 없거나 의존성이 해결된 것 순으로 순서를 정한다.

1.1.4.6. 측정

SMS에서 당일 열차운행계획이 없는 경우 데이터필드에 열차운행계획을 요청하는 메시지를 전송한다.
LIS에서 열차 운행계획 요청 메시지를 수신한다.
LIS는 만일 해당역에 설정된 열차 운행계획ID를 가지는 운행계획 파일이 시스템에 미리 생성되어 있는지 검사한다.
만일 시스템에 미리 운행계획 파일이 생성되어 있지 않다면 '역번호.운행계획ID'를 가지는 운행계획 파일을 생성한다.
LIS에서 운행계획 헤더 정보를 데이터필드에 전송한다.
LIS에서 운행계획 파일의 모든 열차에 대하여 운행계획을 데이터필드에 전송한다.
SMS는 운행계획 헤더가 수신되면 헤더 정보의 메시지를 이용하여 운행계획 파일을 생성한다.
SMS는 운행계획 데이터가 입력되면 순서대로 수신된 운행계획 정보를 저장한다.
LIS의 운행계획 전송이 완료되면 SMS에서 운행계획이 정상적으로 모두 수신되었는지 확인한다.

1.1.4.7. 중단

테스팅을 위해 구동된 응용 프로그램들을 종료시킨다.

1.1.4.8. 재개

테스팅을 재개하기 위하여 각 서브시스템의 응용 프로그램들을 다시 시작한다.
테스트를 위한 데이터는 모두 초기화 시킨다.

1.1.4.9. 중지

해당없음.
1.1.4.10. 종료(마감)
MANTIS와 기록된 테스트 로그를 이용하여 리포트를 작성한다. 테스트 결과 데이터를 다시 형상관리 시스템으로 업데이트 한다. 테스팅을 위해 구동된 응용 프로그램들을 종료시킨다.
1.1.4.11. 비상대처 상황
버그를 보고하기 위하여 MANTIS를 이용한다. 테스트 매니저는 그 정보를 사고 보고서를 작성하는데 사용한다.

* 위의 테스트 시나리오는 IEEE 표준 829항의 테스트 절차 사양서를 따른 것이다.

2. 현장 실 데이터 DB 구축

종합연계시험센터에서 시험을 위해서는 열차스케줄을 비롯한 열차운행정보가 반드시 필요하다. 종합연계시험센터는 자율분산 역제어시스템과 라인관리 시스템, 그리고 라인센터의 기능 시험에 필요한 CTC 에뮬레이션 서버까지 현장 설비에 대한 정보를 공통으로 적용하도록 해야 한다.

종합연계시험센터에서 열차운행정보를 확보하는 방안은 크게 두 가지로 구분할 수 있다. 가장 좋은 방법은 현재 운행중인 CTC의 열차운행정보를 종합연계시험센터에서 직접 실시간으로 받는 방법이다. 다른 한가지 방법은 CTC에 저장된 열차운행실적을 이용하여 시험에 필요한 가상의 시험용 열차운행정보를 가공하는 것이다. 본 연구에서는 후자의 방법을 사용하였다. 열차운행정보를 직접 연결하기 위해서는 보안에 필요한 많은 준비가 필요하기 때문에 향후 실용화 연구 등을 통하여 다시 추진할 예정이다.

현장 설치시험을 위하여 현장 신호설비에 대한 정보를 상호 공유하도록 하기 위해서는 기존에 구로 통합관제센터에서 관리되고 있는 연동 정보나 열차운행계획을 공유할 수 있도록 데이터베이스를 가져와야 한다.

(1) 구로 CTC센터의 데이터베이스 마이그레이션

구로 CTC센터의 데이터베이스는 IBM사의 Informix로 되어 있으며 라인센터의 데이터베이스는 알티베이스사의 Altibase4로 되어 있다. 두 데이터베이스 간에는 일정 부분 상호 호환이 되는 부분도 있으나 일부 데이터는 데이터베이스의 특성에 따라 강제로 형 변환과 데이터베이스 함수를 이용하여야 하는 경우가 발생되었다.

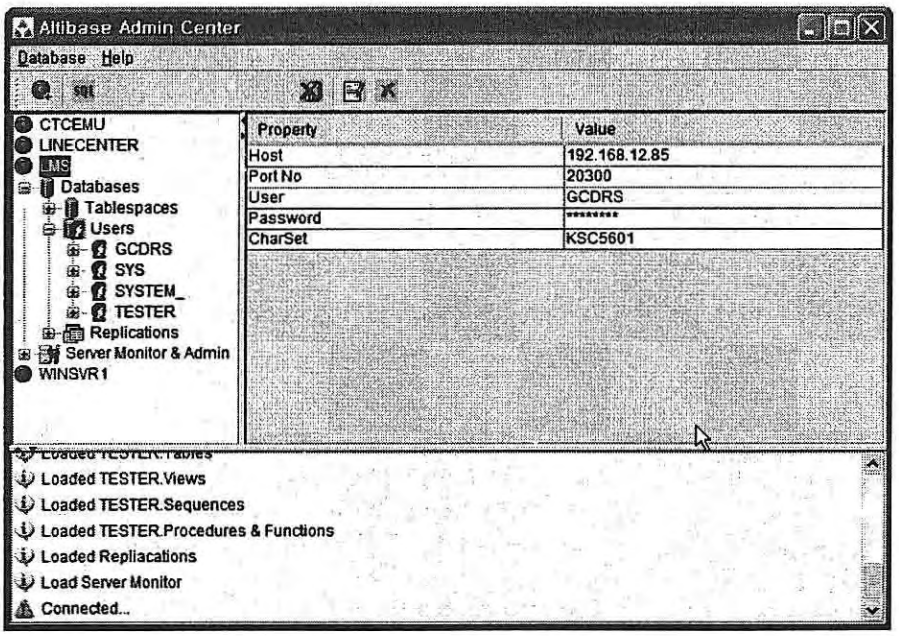


그림 5-2 Altibase Admin TOOL

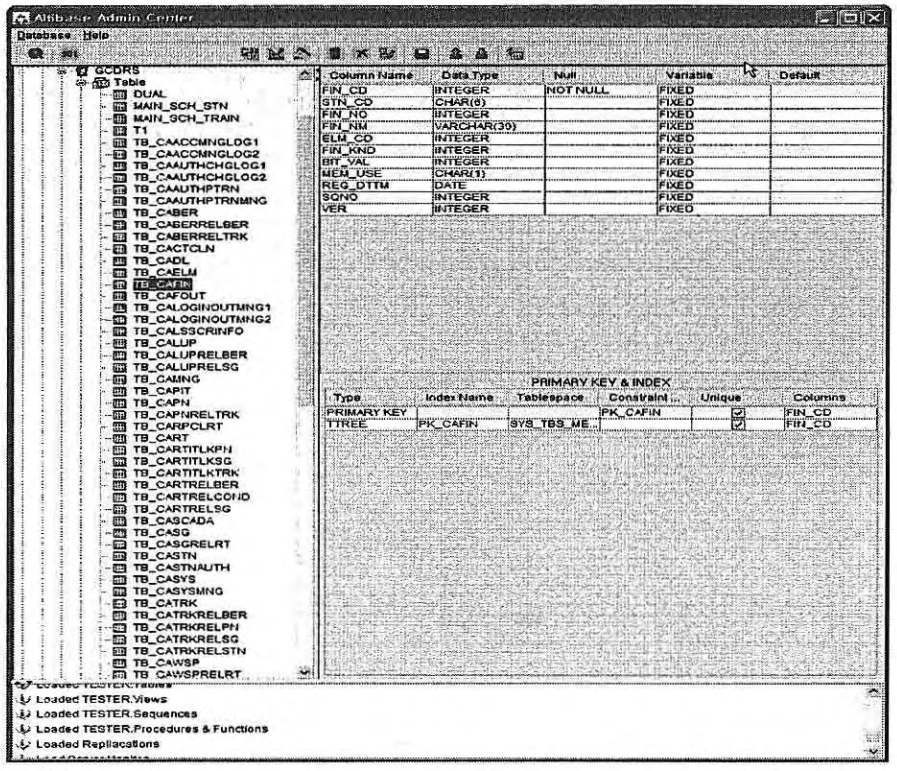


그림 5-3 마이그레이션 된 구로CTC 데이터베이스

(2) CTC 데이터베이스 마이그레이션을 위한 쿼리

(가) 데이터베이스 로드

```
${ALTIBASE_HOME}/bin/iloader -s 127.0.0.1 -u GCDRS -p GCDRS in -f GCDRS_DUAL.fmt -d GCDRS_DUAL.dat -log GCDRS_DUAL.log -bad GCDRS_DUAL.bad
${ALTIBASE_HOME}/bin/iloader -s 127.0.0.1 -u GCDRS -p GCDRS in -f GCDRS_MAIN_SCH_STN.fmt -d GCDRS_MAIN_SCH_STN.dat -log GCDRS_MAIN_SCH_STN.log -bad GCDRS_MAIN_SCH_STN.bad
이하 데이터베이스 테이블 목록 ...
```

(나) 인덱스 갱신

```
${ALTIBASE_HOME}/bin/isql -s 127.0.0.1 -u GCDRS -p GCDRS -f GCDRS_CRT_INDEX.sql
```

```
--#########################
-- GCDRS.TB_CACTCLN
--#########################
alter table TB_CACTCLN add constraint PK_CACTCLN primary key(CTCLN_CD) using index tablespace SYS_TBS_MEMORY;
--#########################
-- GCDRS.TB_CADL
--#########################
alter table TB_CADL add constraint PK_CADL primary key(DL_CD) using index tablespace SYS_TBS_MEMORY;
--#########################
-- GCDRS.TB_CAELM
--#########################
alter table TB_CAELM add constraint PK_CAELM primary key(ELM_CD) using index tablespace SYS_TBS_MEMORY;
이하 데이터베이스 Index ...
```

(다) Foreign Key 구성

```
${ALTIBASE_HOME}/bin/isql -s 127.0.0.1 -u GCDRS -p GCDRS -f GCDRS_CRT_FK.sql
```

```
-- GCDRS.TB_CABERRELBER
alter table TB_CABERRELBER add constraint FK_CABERRELBER_01 foreign key(BER_CD) references GCDRS.TB_CABER(BER_CD);
-- GCDRS.TB_CABERRELTRK
alter table TB_CABERRELTRK add constraint FK_CABERRELTRK_01 foreign key(BER_CD) references GCDRS.TB_CABER(BER_CD);
이하 데이터베이스 Foreign Key ...
```

3. CTC에뮬레이션 서버와 타 시스템간 I/F 개발

CTC에뮬레이션 서버는 구로 통합관제센터에 구축되어 있는 소프트웨어의 기능을 구현하며, 종합연계시험센터에 구축되는 라인정보 서브시스템의 열차 운행계획 정보 전송과 비 ASCS역의 현장 신호설비 상태정보 전송을 위하여 구로 통합관제 CTC서버에는 구축되지 않은 외부 시스템과의 인터페이스를 추가하였다.

(1) 현장 신호설비 상태정보 전송

CTC센터는 현재 CTC화 되어 있는 모든 역의 정보가 입력되므로 이 정보 중에서 시험 시스템에 해당되는 정보만을 선택적으로 전송하도록 한다. 라인정보 서브시스템은 이와 같은 비 ASCS구간에 대한 정보가 CTC 에뮬레이션 서버에서 입력되면 다시 이 정보를 ASCS역이 사용하도록 되어 있는 데이터필드에 전송한다.

ASCS역들은 이 정보 중에서 자신의 인접역에 해당되는 정보를 선택적으로 사용한다.

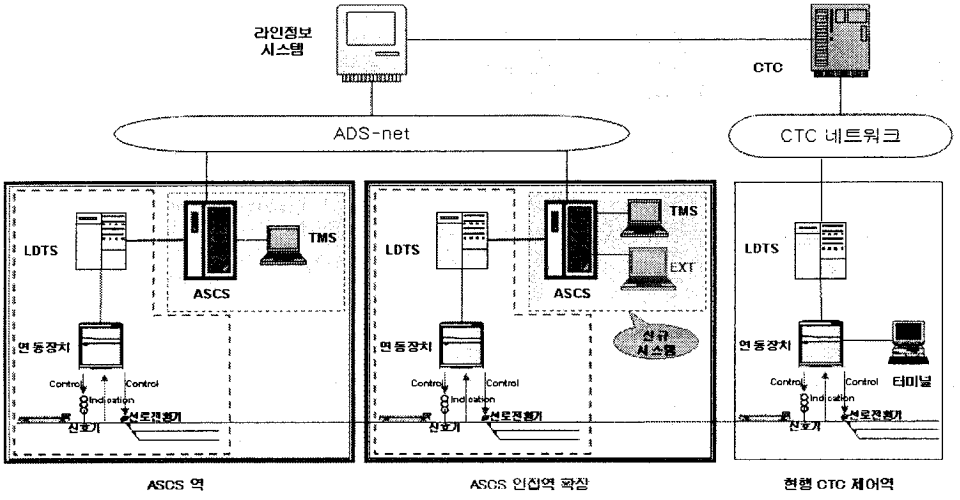


그림 5-4 현장 신호설비 상태정보 전송

(2) 통합관제센터 스케줄 전송

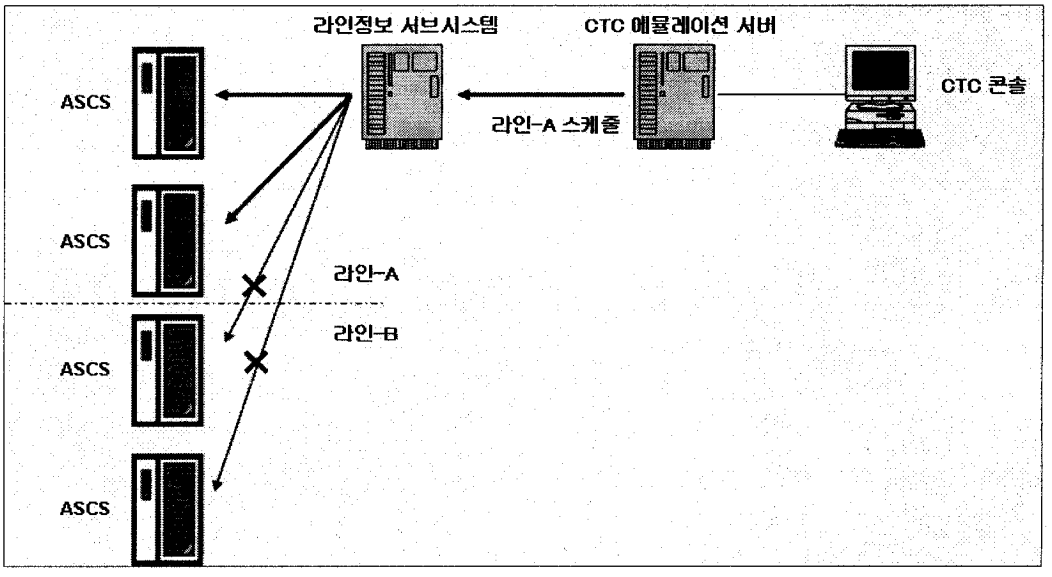


그림 5-5 CTC에뮬레이션 서버의 스케줄 전송

(가) 온라인 스케줄 정보 구조

```
typedef struct
{
    WORD    FuncCode;              /* 메시지 Function Code                */
    BYTE    ConIdx;                /* 콘솔 정보 테이블의 인덱스           */
    BYTE    LnIdx;                 /* 선 정보 테이블의 인덱스             */
    BYTE    TrnNo[TRN_NO_LEN];     /* 열차번호                            */
    BYTE    LnkNo[TRN_NO_LEN];     /* 연계열차번호로 변경                 */
    BYTE    TrnGrd[2];             /* 열차등급(KTX...)                    */
    BYTE    Dir;                   /* 열차방향 0x01:상행 0x02:하행        */
    BYTE    Day;                   /* 0x00:어제, 0x01:오늘, 0x02:내일     */
    BYTE    RunDvCd;   /* 운행구분코드(정기:0x01, 부정기:0x02, 임시:0x03, 현시각:0x09)*/
    BYTE    RunElpsDno;            /* 운행 경과일                         */
    BYTE    StnCnt;                /* 경유역 수                           */
    BYTE    MainRunCd[2];          /* 주운행선코드                        */
    BYTE    Kind[3];               /* 운행차종 (디젤...)                  */
    BYTE    SpeedType[2];          /* 속도종별 (특갑...)                  */
    WORD    StartStnIdx;           /* 시발역인덱스                        */
    BYTE    StartTm[6];            /* 시발시각                            */
    WORD    EndStnIdx;             /* 종착역인덱스                        */
    BYTE    EndTm[6];              /* 종착시간                            */
    BYTE    TrnTypeCd;             /* 열차속성코드(수학여행...)           */
    BYTE    TrnAnxCd;      /* 분할합병구분코드(분할[0x01] 합병[0x02] 순환[0x03])  */
    BYTE    TrnFlag;       /* 회송[0x01] 영업열차[0x02] 직통열차[0x04] 입/출고[0x08] */
```

```
    BYTE    TrnCnt;         /* 차량수                                          */
    BYTE    Suspend;        /* 운휴여부(운행[0x00], 부분운휴[0x01], 전체운휴[0xFF])   */
    WORD    TotalTrnCnt;    /* 여러데이터 전송시 카운터(1대 차량 정보 송신시 0이 셋팅됨)*/
    BYTE    IsResult;  /* 실적여부 없음 : 0 있음 : 1                          */
    BYTE    DirFlag;  /* 열차운행방향변경 플래그                             */
    BYTE    NextMsgFlag; /* 후속메세지 존재여부 플래그 1: 분할/접속메세지 있음    */
    struct {
        WORD    StnIdx;             /* 역 정보 테이블의 인덱스                 */
        BYTE    ArvElpsDno;         /* 도착 경과일                             */
        BYTE    ArvTm[6];           /* 도착 시각 (HHMMSS)                      */
        WORD    ArvRut;             /* 도착 진로                               */
        BYTE    DptElpsDno;         /* 출발 경과일                             */
        BYTE    DptTm[6];           /* 출발 시각 (HHMMSS)                      */
        WORD    DptRut;             /* 출발 진로                               */
        BYTE    RArvElpsDno;        /* 도착 경과일                             */
        BYTE    RltArvTm[6];        /* 실적 도착 시각 (HHMMSS)                 */
        WORD    RltArvRut;          /* 실적 도착 진로                          */
        BYTE    RDptElpsDno;        /* 출발 경과일                             */
        BYTE    RltDptTm[6];        /* 실적 출발 시각 (HHMMSS)                 */
        WORD    RltDptRut;          /* 실적 출발 진로                          */
        BYTE    StnFlag;    /* 실적운휴지연여부 표시(없음[0x00] 운휴[0x01] 지연 [0x02])*/
        BYTE    ResultFlag;  /* 실적정보 0x00:없음 0x01:도착실적 0x02:도착출발실적  */
        BYTE    StpTypeCd[2];       /* 정차구분코드                            */
        BYTE    PlNo;               /* 플랫폼번호                              */
        BYTE    StpTm[6];           /* 정차 시각                               */
        BYTE    DptStpDno;          /* 정차 경과일                             */
        BYTE    RltStpTm[6];        /* 실적 정차시각                           */
        BYTE    StnDir;             /* 역 진행 방향                            */
        BYTE    Dummy;
    } Stn[200];
} TRN_SCH_MOD_INFO;
```

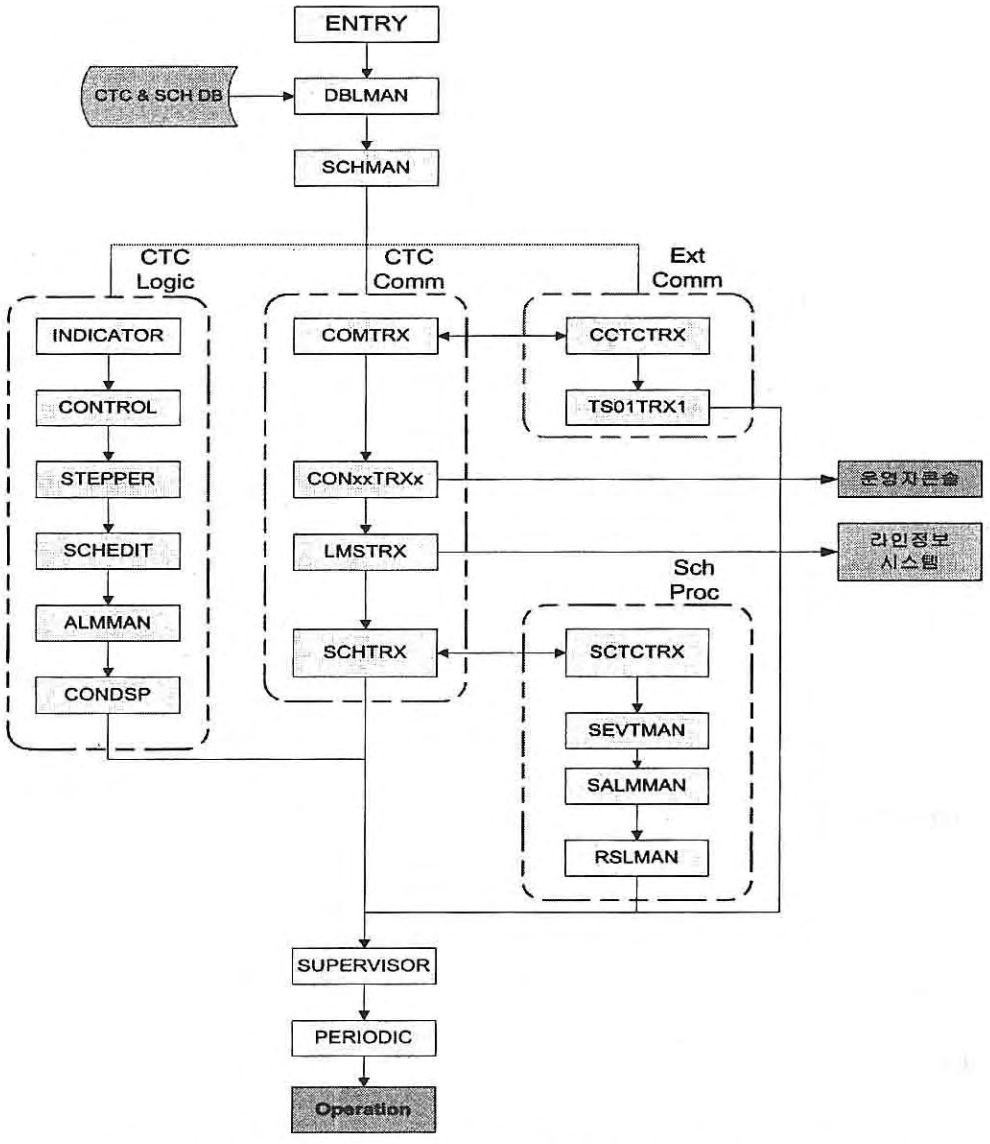


그림 5-6 CTC에뮬레이션 서버의 프로세스 기동 흐름(변경 후)

4. 에뮬레이터 보완 제작

가. 역/이벤트 에뮬레이터

(1) 개요

역 에뮬레이터는 현장의 신호 설비 즉, LDTS의 동작을 시뮬레이션 한다.

역 에뮬레이터가 선로전환기, 궤도회로, 신호기 등의 현장 설비 정보를 주기적으로 ILI로 송신하면 L/S는 이 현장 정보를 수신하여 화면에 표시한다. 또한 역 에뮬레이터는 L/S로부터 수동 제어 명령을 수신하여 현재 연동 조건을 만족하는지 체

크하고 이를 반영한 후 갱신된 정보를 ILI로 재송신 하는 역할을 한다.

(2) SEMUL 프로그램

(가) 기능

① 표시

- 선로전환기 상태 표시

선로전환기의 현재 상태(정위, 반위, 고장, Locking)를 역 에뮬레이터에 표시한다.

- 신호기 상태 표시

신호기의 현재 상태(진행, 고장, 선별등, 접근, 자동)를 역 에뮬레이터에 표시한다.

- 궤도회로 상태 표시

현재 궤도회로의 점유 유무 상태를 역 에뮬레이터에 표시한다.

- 메시지 송수신 표시

메시지의 송수신을 역 에뮬레이터에 표시한다.

(나) 제어

① 선로전환기 제어

- RTPS

RTPS로부터 선로전환기의 변경 메시지를 수신하여 원하는 선로전환기의 상태를 변경한다.

- 사용자

사용자로부터 GUI 환경을 통해 원하는 신호기의 상태를 변경할 수 있다.

② 신호기 제어

- RTPS

RTPS로부터 신호기의 상태 변경 메시지를 수신하여 원하는 신호기의 상태를 변경한다.

- 사용자

사용자로부터 사용자 환경을 통해 원하는 신호기의 상태를 변경 할 수 있다.

③ 궤도회로 점유, 해제 설정

- 다중열차운행시뮬레이터

다중열차운행시뮬레이터를 통해 궤도회로의 점유, 해제 설정을 할 수 있다.

- 사용자

L/S에 궤도회로가 정확한 위치에 점유 되었는지 확인하기 위해 사용자로부터 GUI환경을 통해 궤도 회로 점유, 해제 설정을 할 수 있다.

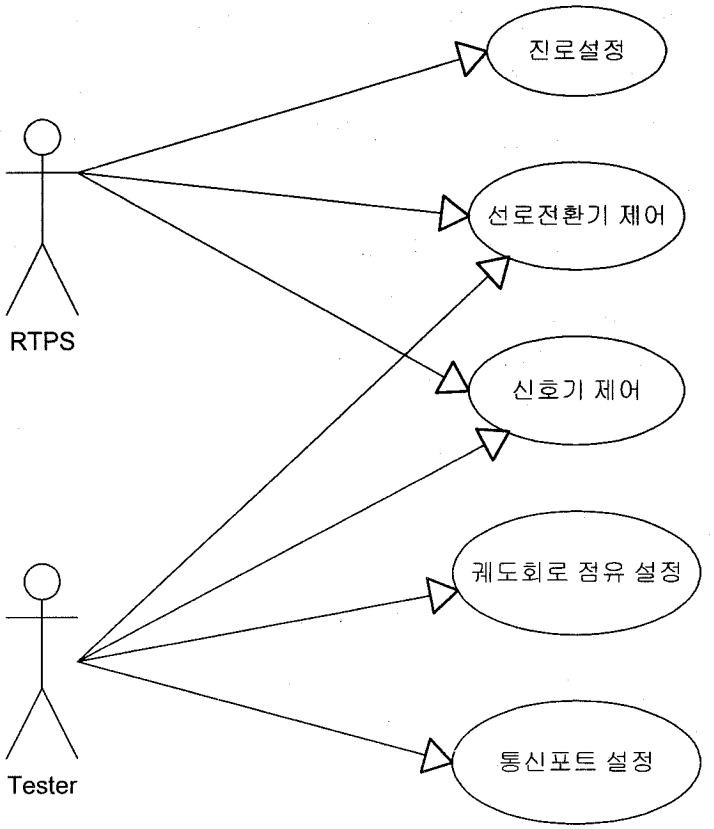


그림 5-7 MTMS 유즈케이스 다이어그램

(3) 유즈케이스

(가) 액터

① RTPS

역 에뮬레이터 시스템과 직접적으로 통신을 하며, 현장 설비의 제어 메시지를 송수신 한다.

② Tester

역 에뮬레이터를 사용하는 사용자로서 선로전환기, 신호기, 궤도회로를 테스트 한다.

(나) 설명

① 선로전환기 제어

- RTPS 또는 Tester로부터 제어 이벤트를 수신하여 선로전환기의 상태를 변경한다.

② 신호기 제어

- RTPS 또는 Tester로부터 제어 이벤트를 수신하여 신호기의 상태를 변경한다.

③ 궤도회로 점유 해제 설정

- Tester로부터 제어 이벤트를 수신하여 궤도회로의 상태를 변경한다.

④ 진로 설정 :

- RTPS로부터 제어 메시지를 수신하여 진로를 설정 하거나 해제 한다.

⑤ 포트 설정

- RTPS와 연결을 위해 포트를 설정한다.

(4) 구조

(가) 시스템 경계

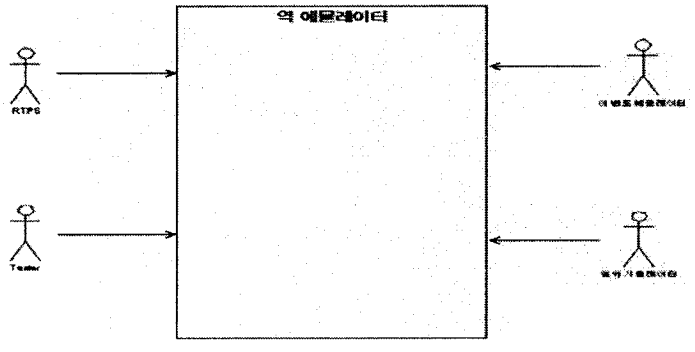


그림 5-8 역 에뮬레이터 시스템 경계

(나) 논리적 뷰

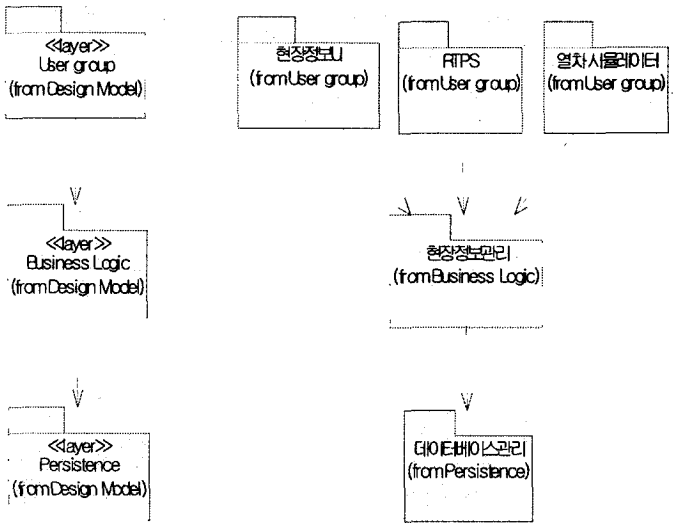


그림 5-9 역 에뮬레이터 로직컬 뷰

(다) 컴포넌트 뷰

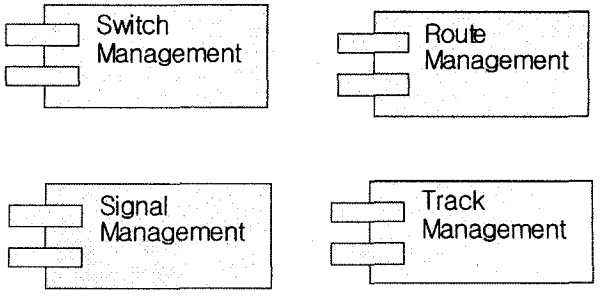


그림 5-10 역 에뮬레이터 컴포넌트 뷰

(라) 컴포넌트 정의

표 5-1 역 에뮬레이터 컴포넌트 정의

컴포넌트 명	유즈케이스 명	구성 요소
Switch Management	선로전환기 제어	선로전환기 변경 관리
Signal Management	신호기 제어	신호기 변경 관리
Track Management	궤도회로 점유 해제 설정	궤도회로 상태 관리
Route Management	진로 설정	진로 설정 관리

(마) 물리적 뷰

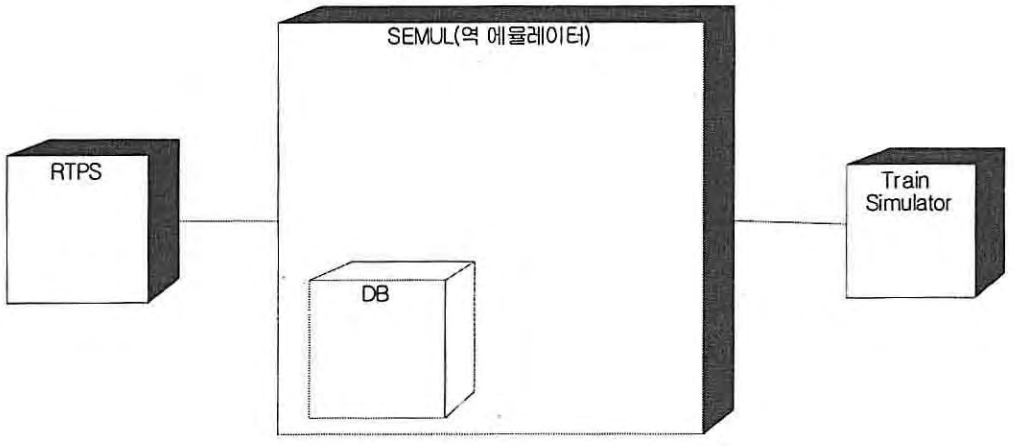


그림 5-11 역 에뮬레이터 물리적 뷰

(5) 사용자 인터페이스

(가) 전체 화면

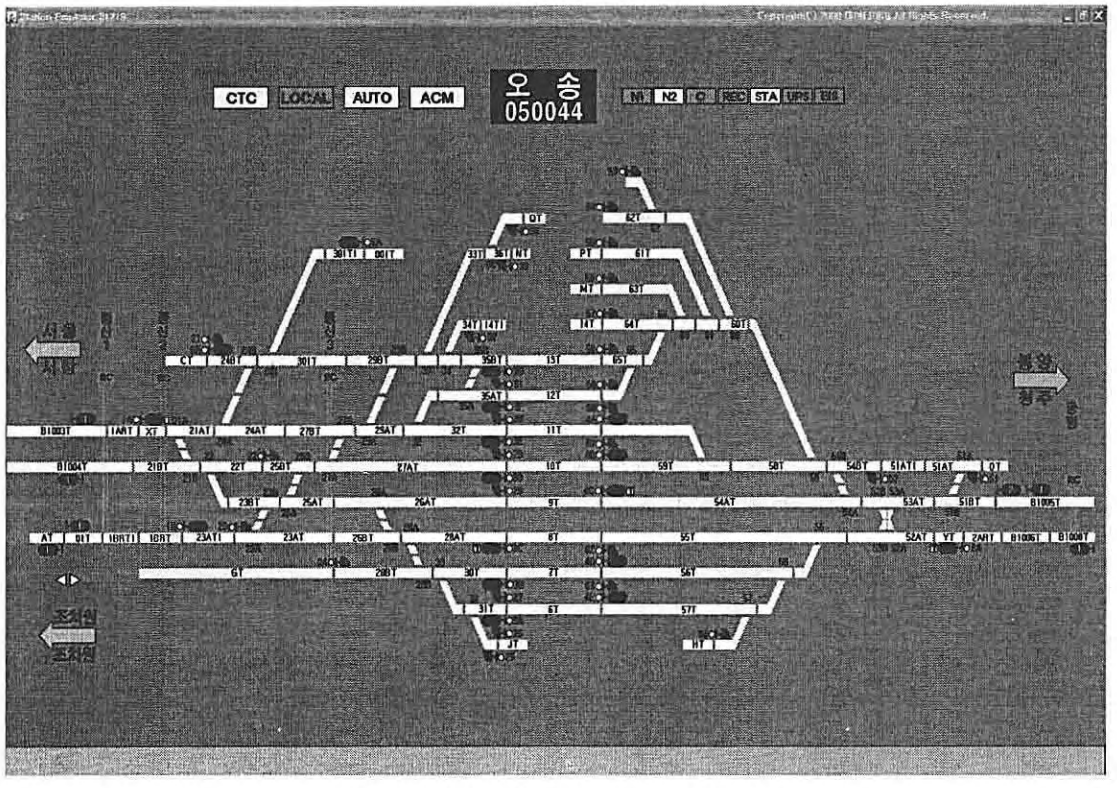
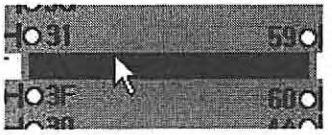


그림 5-12 역 에뮬레이터 전체 화면

(나) 현장 제어

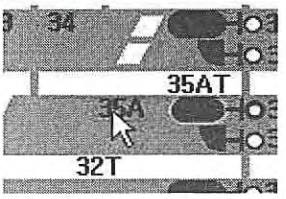
① 궤도회로 제어

표 5-2 궤도회로 정보 송신

설명	화면
1. 점유 시키고자 하는 궤도회로 위에 마우스 커서를 이동 시킨 후 마우스 왼쪽 버튼 또는 오른쪽 버튼을 클릭 한다.	
2. 마우스 왼쪽 버튼을 클릭하면 선택된 궤도회로가 점유로 표시 되고 제어 정보는 ILI로 송신된다. 3. 점유를 해제 하려면 점유된 궤도회로 위에 마우스 커서를 이동 시킨 후 마우스 왼쪽 버튼을 클릭 한다.	

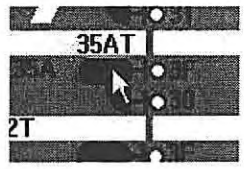
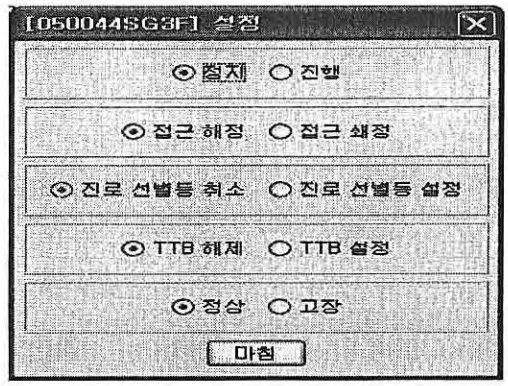
② 선로전환기 제어

표 5-3 선로전환기 정보 송신

설명	화면
1. 제어 하고자 하는 선로전환기 명위에 마우스 커서를 이동 시킨 후 마우스 오른쪽 버튼을 클릭 한다.	
2. 제어 하고자 하는 라디오 버튼을 선택한 후 <마침> 버튼을 클릭한다.	

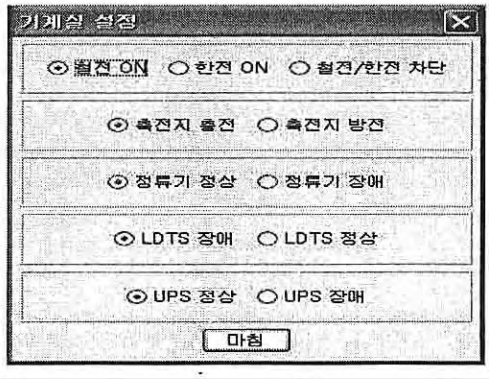
③ 신호기 제어

표 5-4 신호기 정보 송신

설명	화면
1. 제어 하고자 하는 신호기 위에 마우스 커서를 이동 시킨 후 마우스 오른쪽 버튼을 클릭 한다.	
2. 제어 하고자 하는 라디오 버튼을 선택 한 후 <마침> 버튼을 클릭한다.	

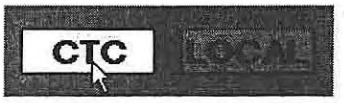
④ 기계실 설정

표 5-5 기계실 설정

설명	화면
1. 기계실 표시 위에 마우스 커서를 이동 시킨 후 마우스 오른쪽 버튼을 클릭 한다.	
2. 제어 하고자 하는 라디오 버튼을 선택 한 후 <마침> 버튼을 클릭 한다.	

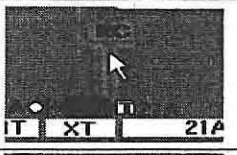
⑤ 역 모드 설정

표 5-6 역 모드 설정

설명	화면
1. 역 모드 위에 마우스 커서를 이동 시킨 후 마우스 오른쪽 버튼을 클릭 한다.	
2. 제어 하고자 모드를 선택 한 후 <마침> 버튼을 클릭한다.	

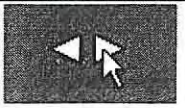
⑥ 건널목 설정

표 5-7 역 모드 설정

설명	화면
1. 제어 하고자 하는 건널목 위에 마우스 커서를 이동 시킨 후 마우스 오른쪽 버튼을 클릭한다.	
2. 제어 하고자 모드를 선택 한 후 <마침> 버튼을 클릭한다.	

⑦ 폐색표시 설정

표 5-8 폐색 표시 설정

설명	화면
1. 패색 표시 위에 마우스 커서를 이동 시킨 후 마우스 오른쪽 버튼을 클릭한다.	
2. 제어하고자 하는 라디오 버튼을 선택한 후 <마침> 버튼을 클릭한다.	

5. CTC L/S 모니터링 콘솔 개발

가. 개요

CTC 에뮬레이터 서버는 CTC 서버로 입력되는 현장 데이터에 대한 처리를 수행하며 CTC L/S 모니터링 콘솔은 CTC 에뮬레이터에서 발생되는 현장 신호설비에 대한 상태정보를 표시한다.

CTC L/S 모니터링 콘솔은 구로 통합관제센터에서 관리되고 있는 CTC L/S MAP 파일과 데이터 포맷을 그대로 이용할 수 있도록 제작되었다. 구로 통합관제센터에서 작성된 L/S MAP 데이터는 Dataview라는 상용 Drawing Tool로 작성되었으므로 이를 위해서 해당 Tool을 사용할 수 있는 라이센스를 이용한다.

나. 현장 설치시험 대상역 L/S

현장 설치시험 대상역이 처음 과제 시작과 달리 오송, 청주, 오근장역으로 선정되었으므로 구로 통합관제센터에서 작성된 Dataview Map을 이용하여 모니터링 콘솔에 L/S화면을 표출하였다. CTC 모니터링 콘솔 L/S에서는 대상 역의 변경이 발생하더라도 구로 통합관제센터에서 MAP을 수정하고 이 MAP을 그대로 사용할 수 있도록 프로그램이 작성되어 있다.

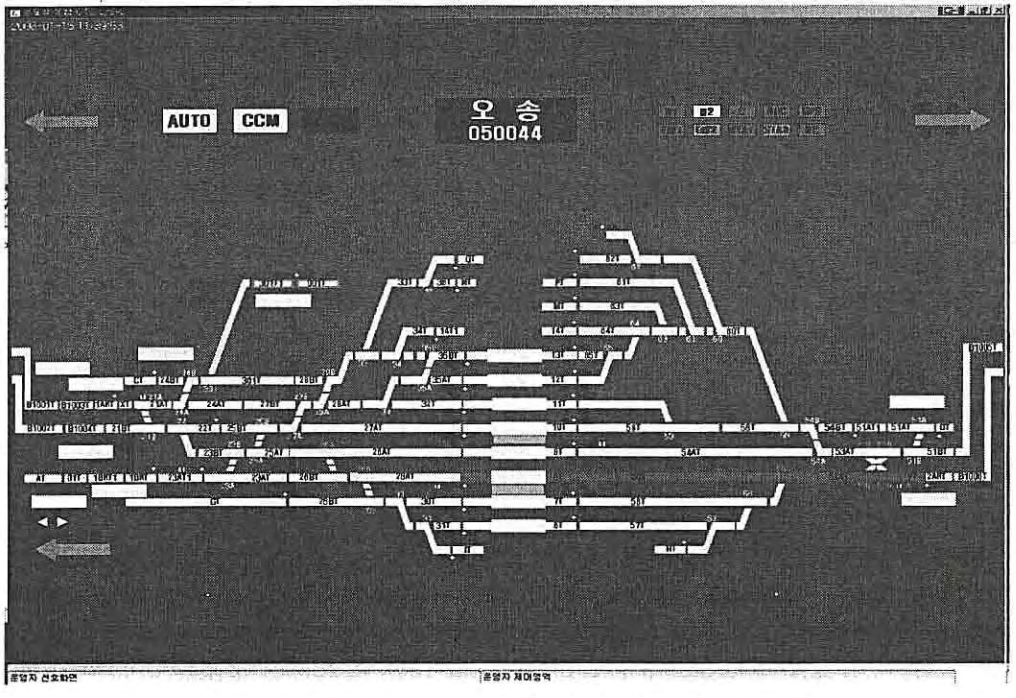


그림 5-13 CTC L/S 모니터링-충북선 오송역

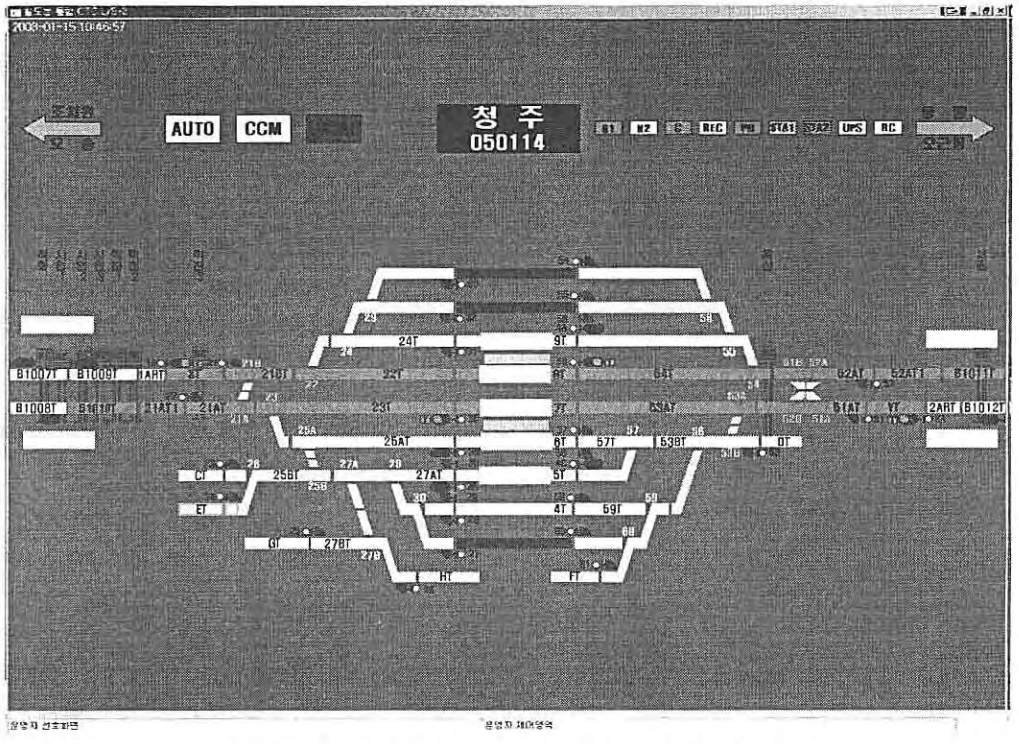


그림 5-14 CTC L/S 모니터링-충북선 청주역

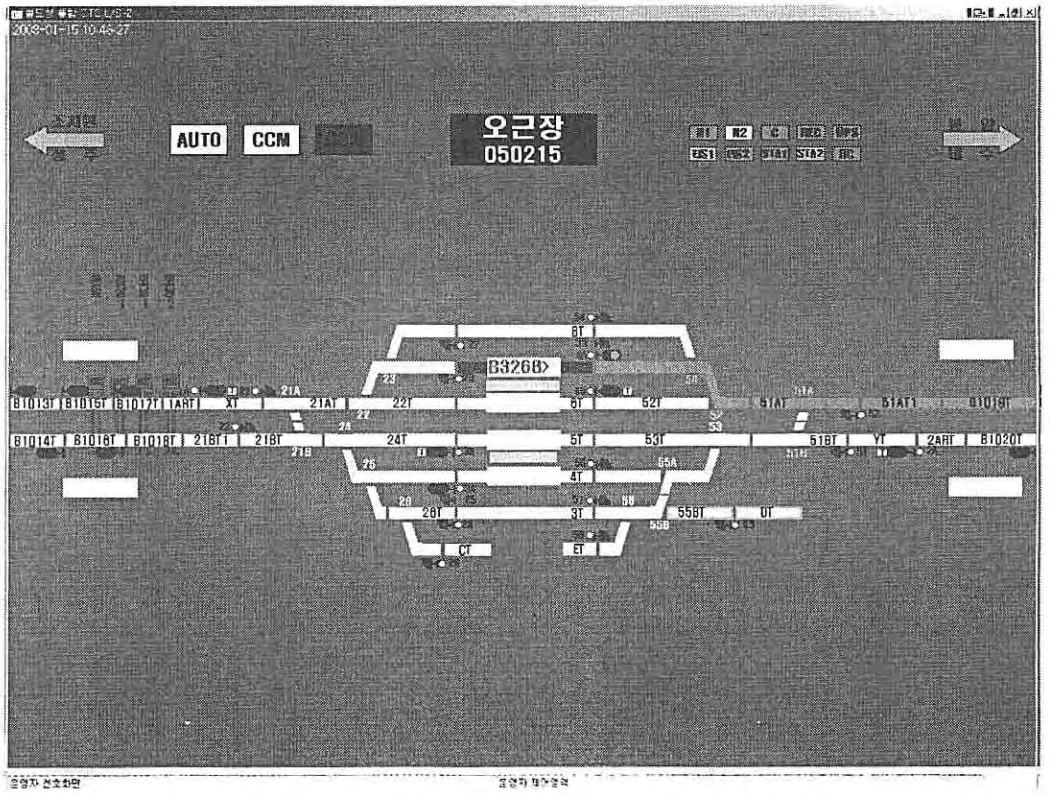


그림 5-15 CTC L/S 모니터링-충북선 오송역

2절 종합연계시험센터 설치 및 시험

1. 종합연계시험센터 설치

가. 설치 개요

종합연계시험센터는 자율분산 역제어시스템의 단계적 구축 및 향후 현장 설치시험을 위한 센터설비의 기능 설계 및 센터설비와 역설비의 상호 협조체제 및 병행운전을 위하여 개발된 시스템이다. 종합연계시험센터는 다음과 같은 개발 과정을 통해 구축되었다.

- 시험센터 설계
- 운행관리, 열차스케줄, 통신 기능 설계 및 구현
- 인터페이스 기능 구현
- 다중열차운행시뮬레이터 개발
- 자율분산 시스템과의 병행운전 및 인터페이스 기능 구현

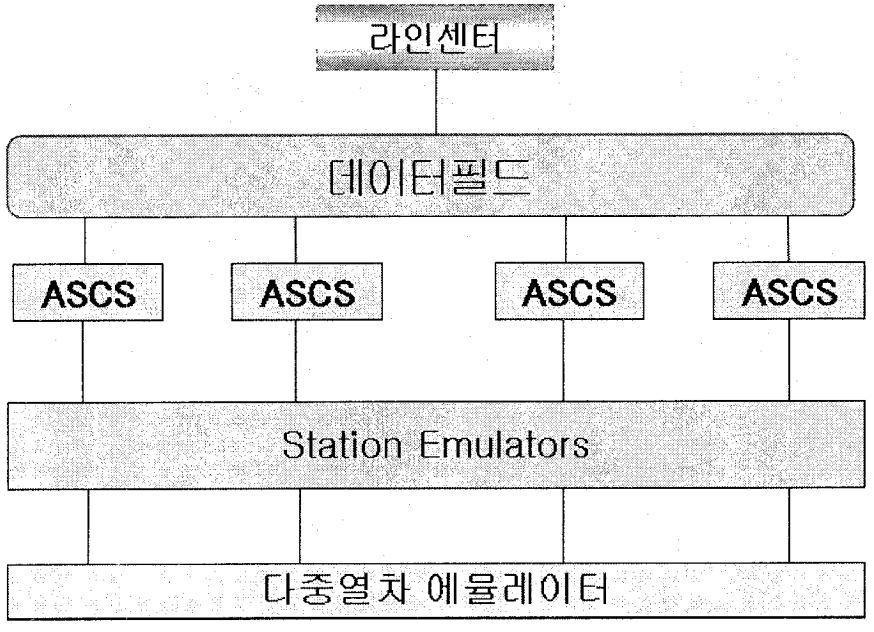


그림 5-16 종합연계시험센터의 논리적 구성

나. 물리적 구성

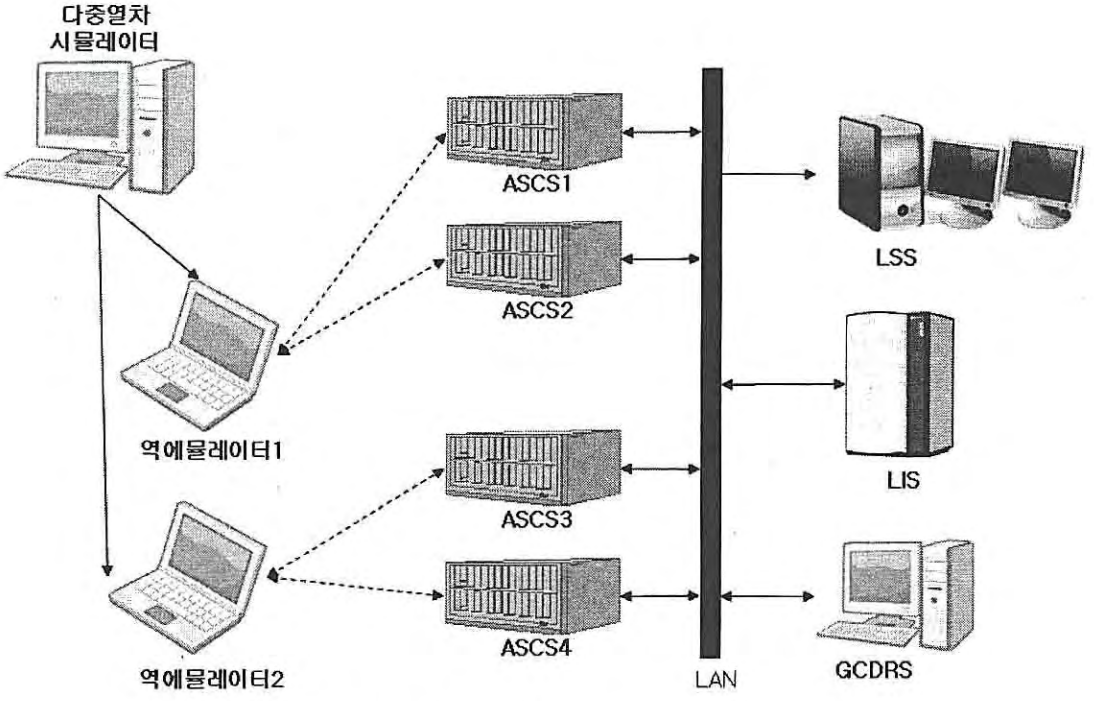


그림 5-17 종합연계시험센터의 물리적 구성



그림 5-18 종합연계시험센터 설치 사진(1)

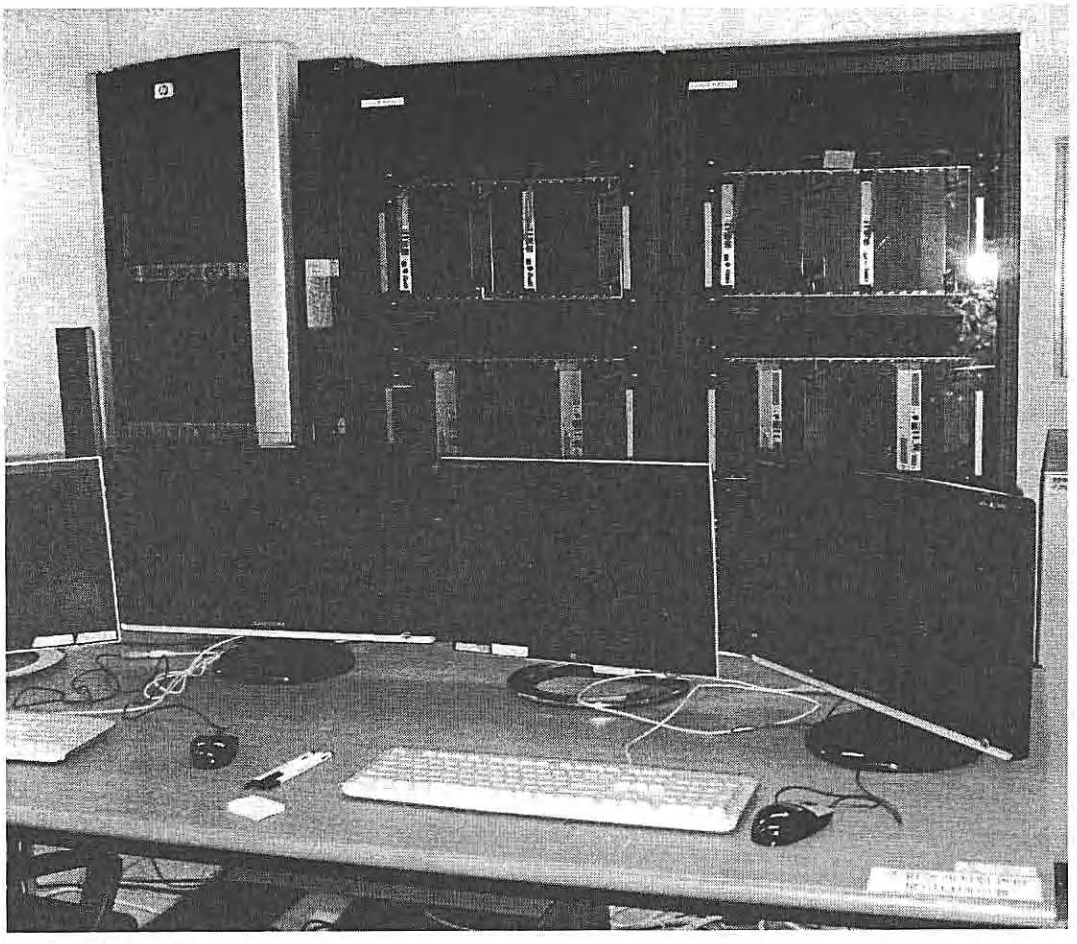


그림 5-19 종합연계시험센터 설치 사진(2)

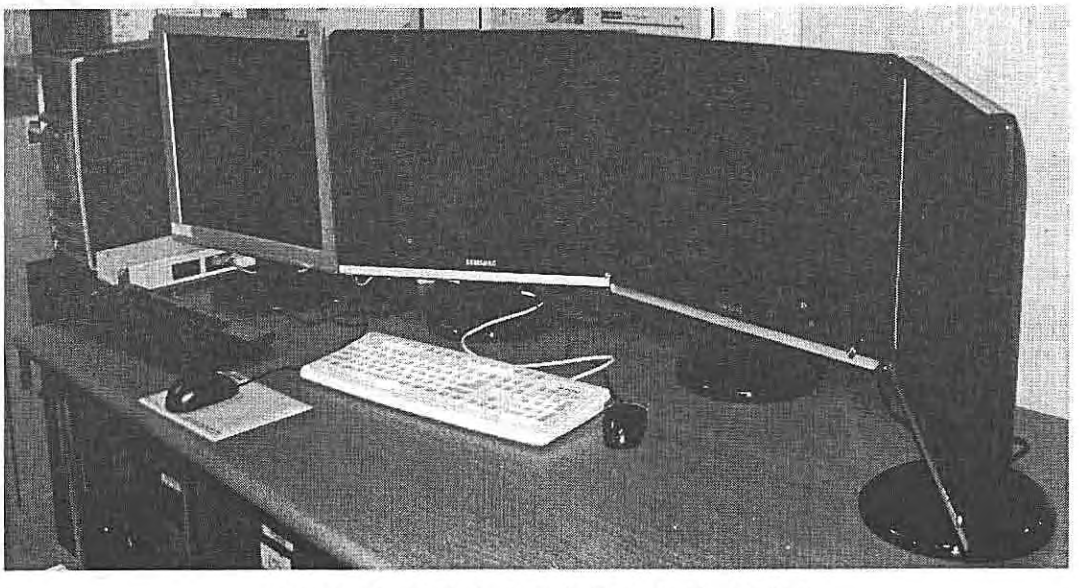


그림 5-20 종합연계시험센터 설치 사진(3)

다. 데이터필드의 구성

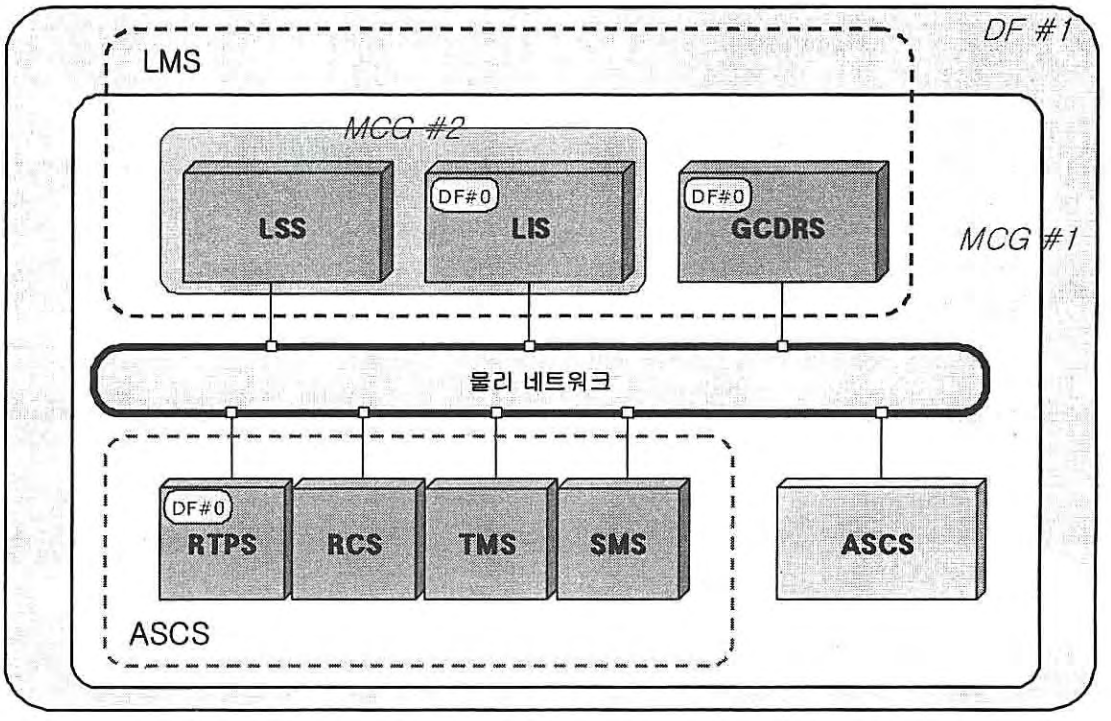


그림 5-21 종합연계시험센터의 데이터필드 구성

종합연계시험센터를 구성하는 시스템의 종류와 수량이 많지 않으므로 종합연계시험센터는 하나의 데이터필드로 구성하였으며, 라인센터 내의 서브시스템간 인터페이스가 역 ASCS의 자율분산관리자에 부하를 주지 않기 위하여 멀티캐스트 그룹이 하나 더 추가되었다.

다중열차운행시뮬레이터는 철도 레일을 따라 운행하는 열차의 이동에 대한 시뮬레이션을 하는 시스템으로 현장 신호설비와의 인터페이스를 위하여 역에뮬레이터와 시리얼로 인터페이스되며, 역 신호설비의 상태에 따라 열차의 이동 방향 및 주행 여부를 결정하기 위하여 자율분산관리자가 이용된다. 다중열차운행시뮬레이터에 사용되는 자율분산관리자는 라인모니터링에 사용된 자율분산관리자를 사용하며, 데이터필드는 1번, 멀티캐스트 그룹은 1번을 사용하고, CC는 ADM 프로토콜에 따른다.

* CC 1 : 역 신호설비 상태정보 표시

* CC 5 : 표준시간

라. 종합연계시험센터 시험 내용

종합연계시험센터의 시험을 위하여 다중열차운행시뮬레이터, 라인센터(LIS, LSS), ASCS(RTPS, RCS, TMS, SMS)를 구동하였다.

(1) 다중열차운행시뮬레이터와 역 에뮬레이터 I/F 시험

다중열차운행시뮬레이터는 현장 신호설비의 상태정보에 따라 역 에뮬레이터에 궤도점유정보를 전송한다. 궤도점유 정보의 전송은 UDP(User Datagram Protocol) 멀티캐스트를 사용하여 네트워크에 연결되어 있는 어떤 지점에서도 다중열차운행시뮬레이터의 궤도점유 정보를 확인할 수 있도록 구성하였다. 시험에는 하나의 다중열차운행시뮬레이터와 하나의 역 에뮬레이터가 사용되었다.

다중열차운행시뮬레이터를 ASCS가 설치되지 않은 시스템에서 구동한다. 다중열차운행시뮬레이터의 구동을 위해서는 다중열차운행시뮬레이터가 필요하므로 다중열차운행시뮬레이터 구동 이전에 다중열차운행시뮬레이터를 구동한다.

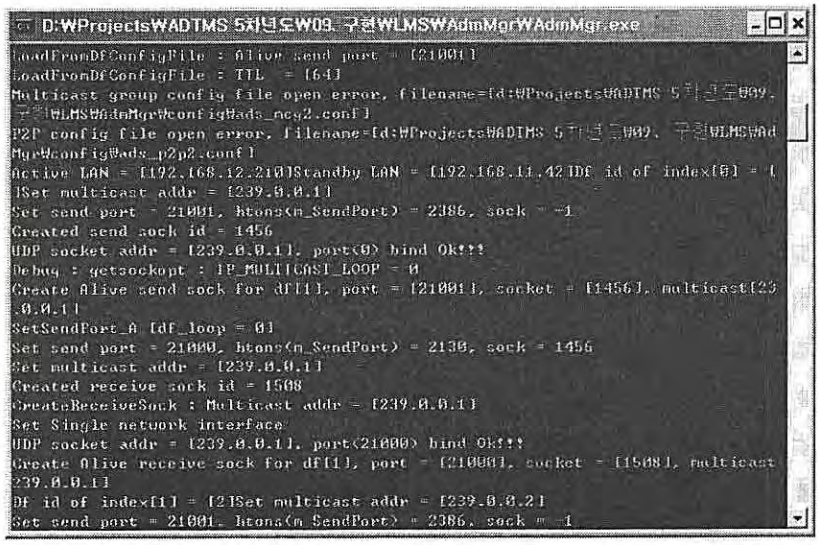


그림 5-22 다중열차운행시뮬레이터를 위한 ADM 구동

시뮬레이터가 구동되는 PC에서 ADM이 구동된 후 다중열차운행시뮬레이터를 시작한다.

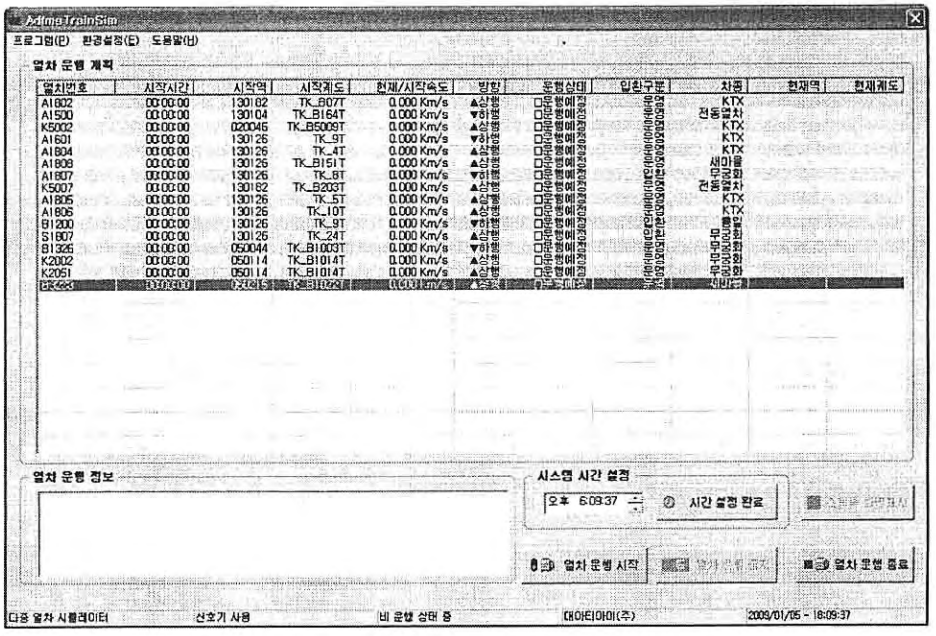


그림 5-23 다중열차운행시뮬레이터 구동

다중열차운행시뮬레이터에서 사용되는 열차스케줄은 시험 환경에 따라 변경되어야 하므로 편집툴을 제공하여 현재 시각에 필요한 스케줄을 편집하여 사용한다.

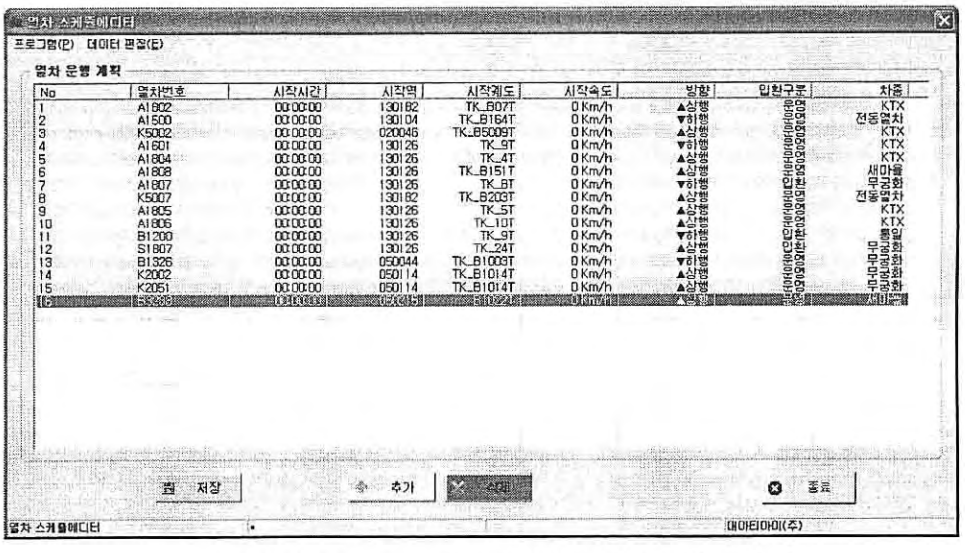


그림 5-24 다중열차운행시뮬레이터-열차편집

다중열차에서 현재의 시험에 해당되는 열차의 운행을 시작하면 다중열차운행시뮬레이터의 궤도점유 정보는 역에뮬레이터에 전송되어 역에뮬레이터에서 ASCS로 궤

도 점유를 전송하게 된다.

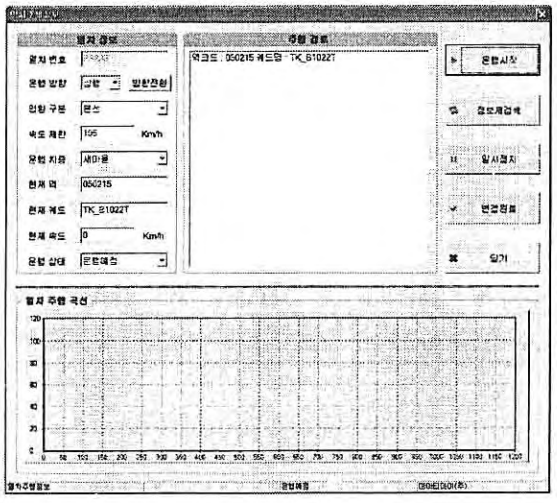


그림 5-25 SIM-열차운행시작

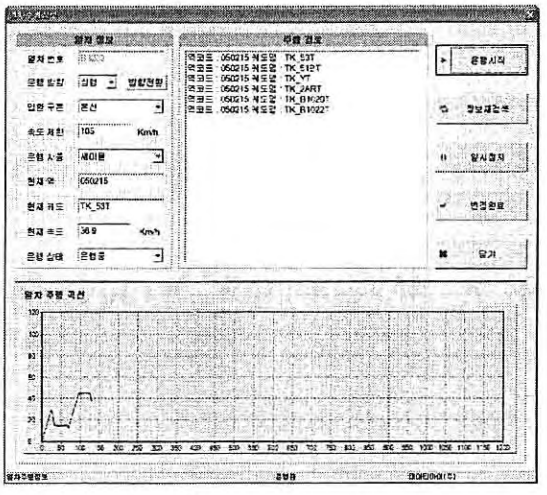


그림 5-26 SIM-열차주행

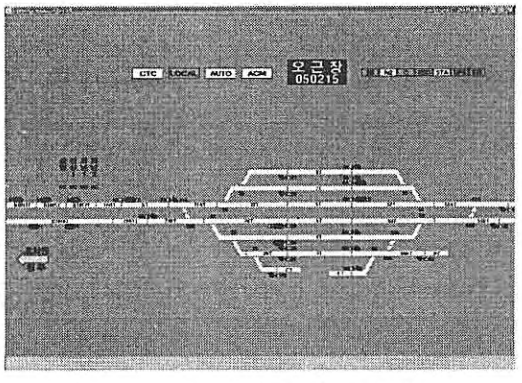


그림 5-27 역에뮬레이터 실행

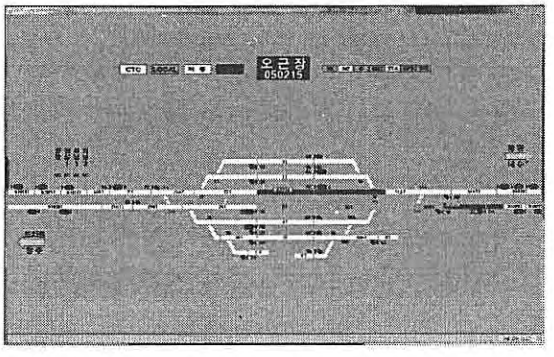


그림 5-28 역에뮬레이터 정보처리

(2) 라인정보 서브시스템의 구동

라인정보 서브시스템은 자율분산관리자와 열차 운행계획관리를 위한 프로세스 및 라인센터의 열차추적을 위한 프로세스, 그리고 비ASCS역에 대한 정보 제공을 위한 프로세스들이 구동된다. LIS는 상용데이터베이스를 이용하므로 LIS의 응용프로그램 구동 이전에 Altibase와 MySQL이 구동되어 있는지 확인되어야 한다.

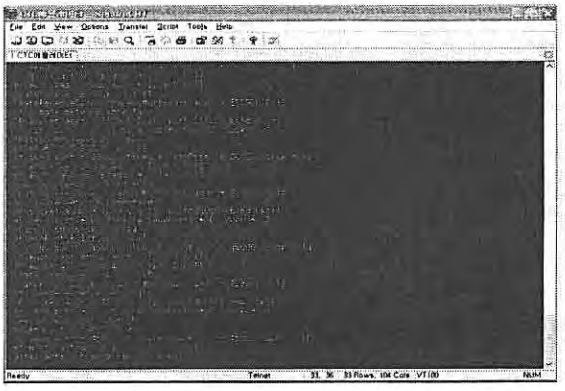


그림 5-29 LIS-자율분산관리자 시작

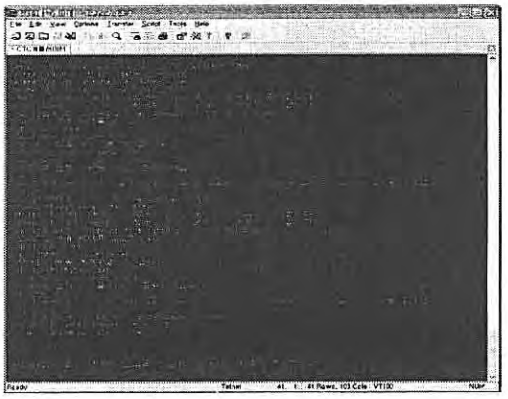


그림 5-30 LIS-스케줄관리자 시작

LIS의 프로세스가 모두 시작된 경우 다음과 같은 정보를 확인할 수 있다.

```
[adtms:adtms:/home/adtms/src/exec] ps -ef | grep adtms
    adtms 11549     1  0 16:30:36 ?        0:00 FieldIndD
    adtms 11545     1  0 16:30:12 ?        0:00 AdmMgrD
    adtms 11551     1  0 16:30:42 ?        0:00 TrainTrkD
    adtms 11553     1  0 16:30:47 ?        0:00 SchMngD
    adtms 11565 11341 65 16:31:09 pts/tc    0:00 AdsStation
[adtms:adtms:/home/adtms/src/exec]
```

(3) 라인정보 서브시스템의 본선스케줄 전송

LIS에서 관리되는 오프라인 본선스케줄은 ASCS의 서브시스템인 SMS에 요청에 따라 ASCS로 전송된다.

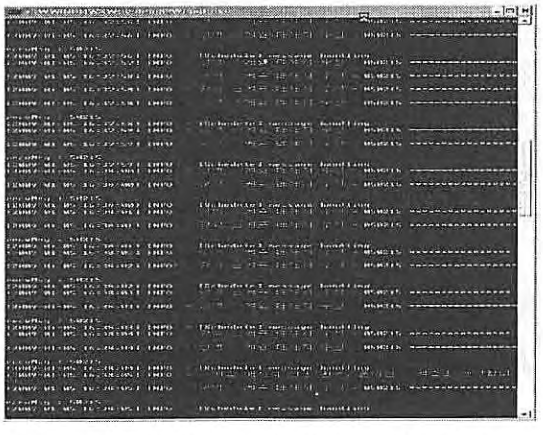


그림 5-31 SMS-오프라인스케줄수신

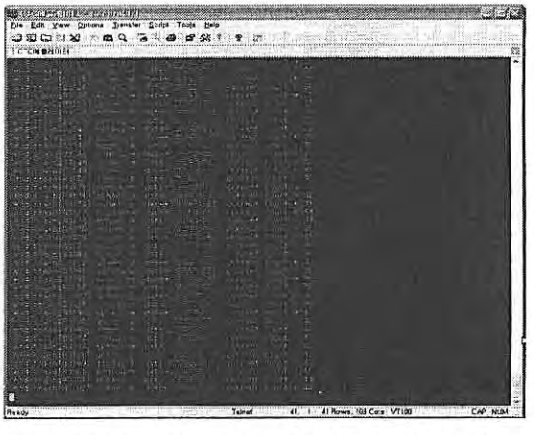


그림 5-32 LIS-오프라인스케줄 전송

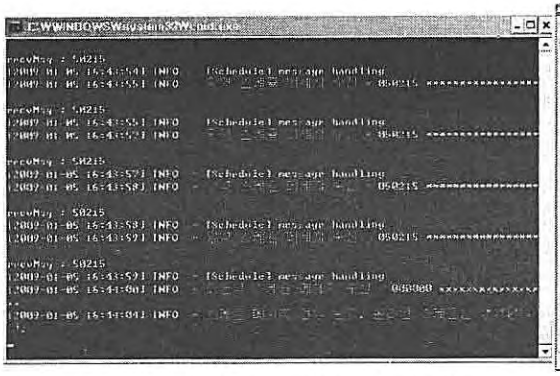


그림 5-33 오프라인스케줄 수신완료

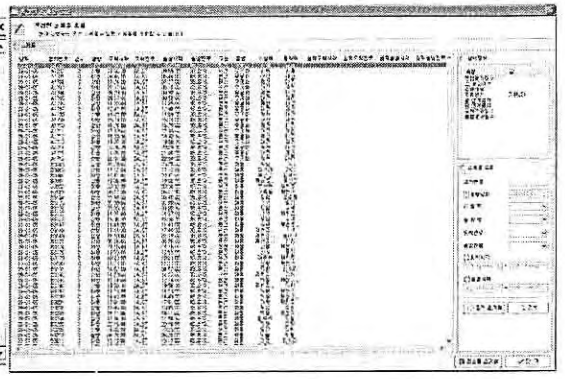


그림 5-34 SMS-수신된 본선스케줄표시

운영자가 LSS의 MMI를 이용하여 온라인 열차스케줄을 수정하게 되면, LIS는 변경된 스케줄 정보를 데이터필드에 전송한다. ASCS의 서브시스템은 SMS는 변경된 온라인 본선스케줄을 수신하게 되면 해당 ASCS의 온라인 스케줄을 변경한다.

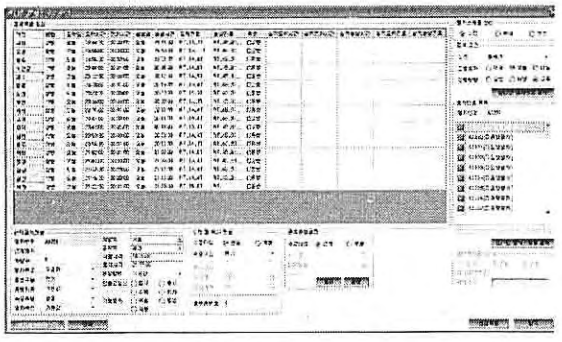


그림 5-35 LSS-온라인스케줄 관리

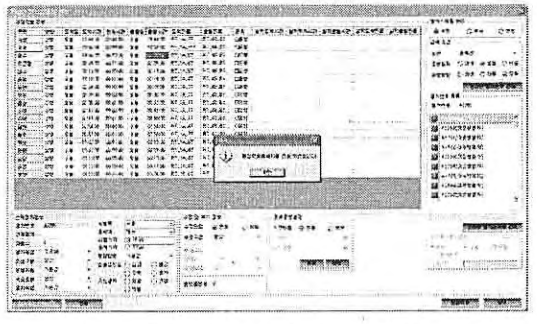


그림 5-36 LIS-온라인스케줄 변경

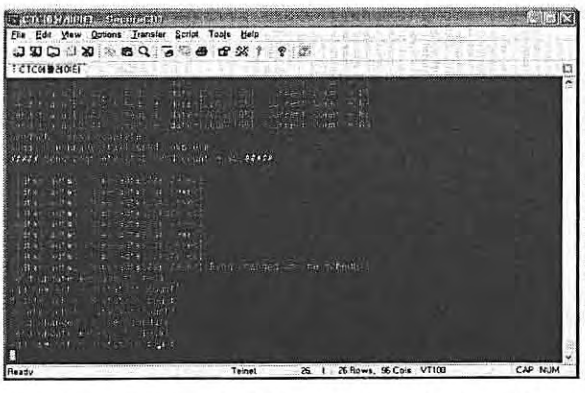


그림 5-37 LIS-온라인스케줄 변경정보 전송

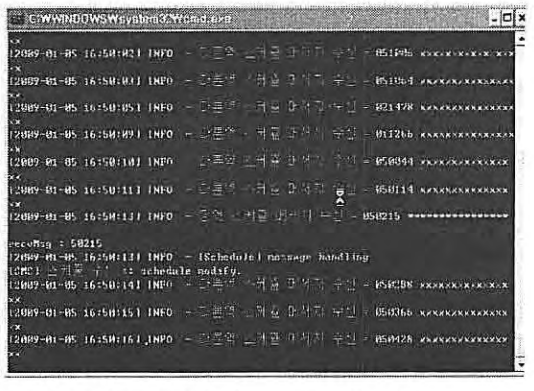


그림 5-38 온라인스케줄 수신완료

(4) ASCS 표시 및 제어 시험

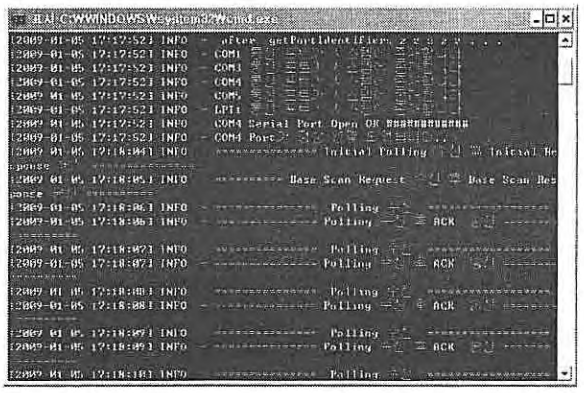


그림 5-39 역에뮬레이터 구동

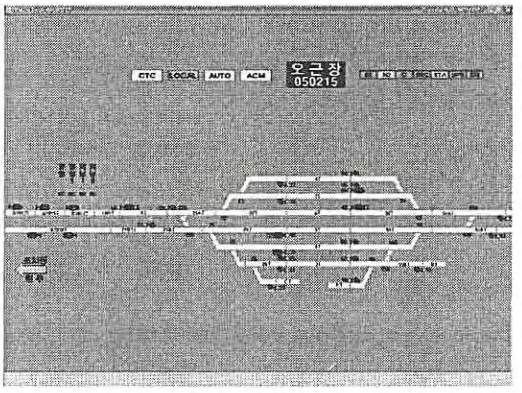


그림 5-40 역에뮬레이터 실행화면

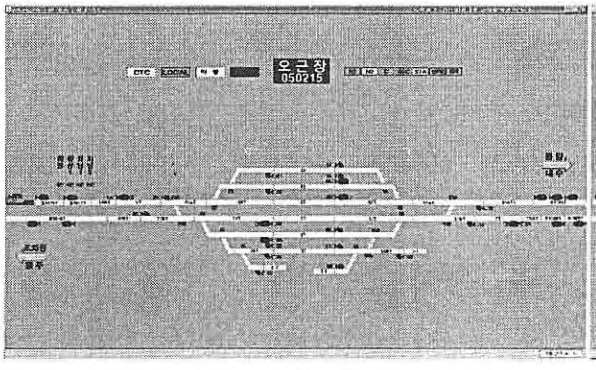


그림 5-41 ASCS 열차 역 진입

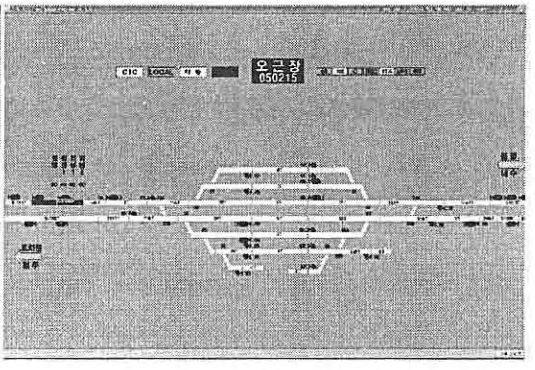


그림 5-42 열차 주행

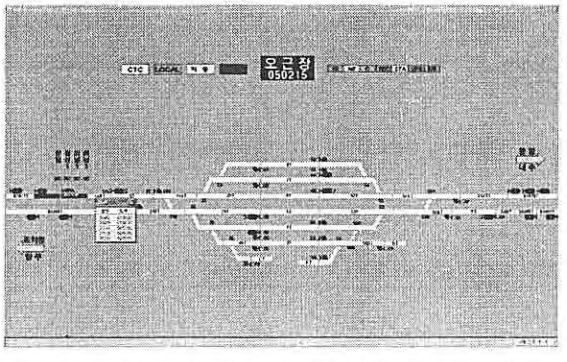


그림 5-43 역 진입예정열차 표시

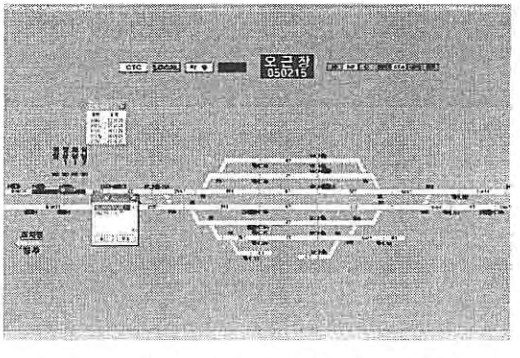


그림 5-44 수동진로제어

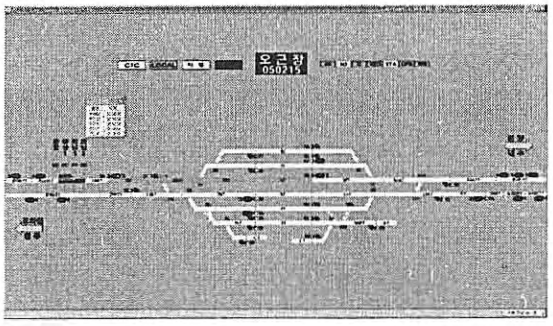


그림 5-45 진입진로 구성

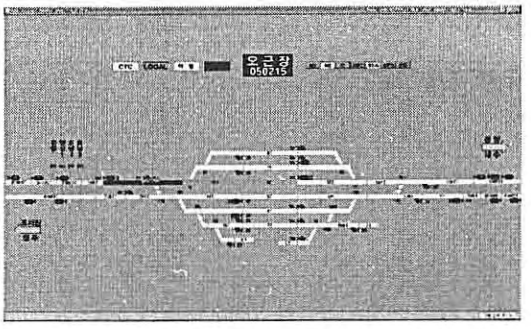


그림 5-46 열차 구내 진입

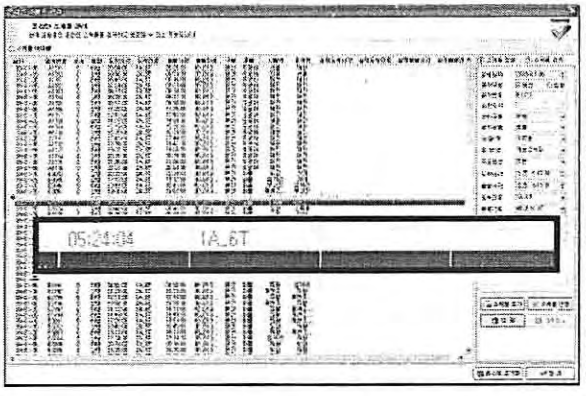


그림 5-47 역 도착실적 기록

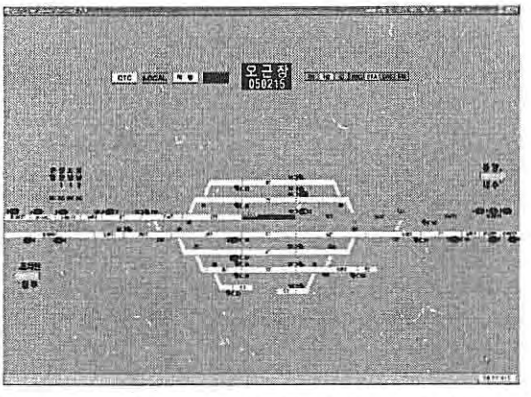


그림 5-48 역 출발진로 설정

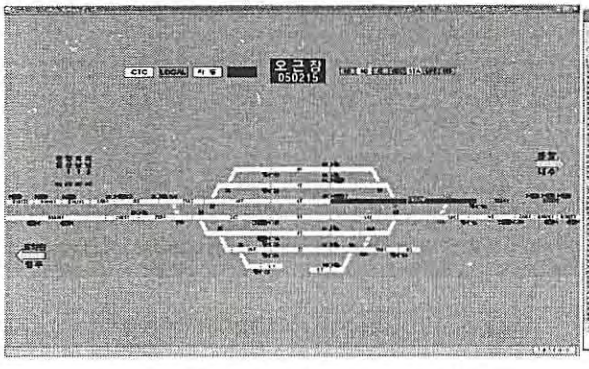


그림 5-49 역 장내 출발

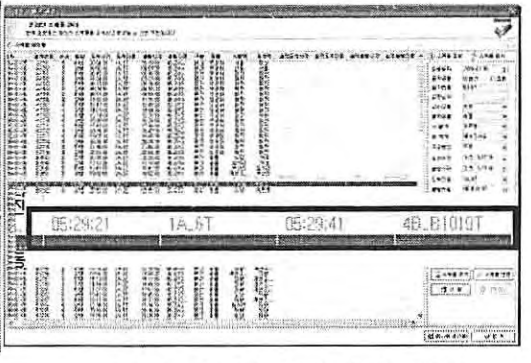


그림 5-50 역 출발실적 기록

역에뮬레이터에서 생성되는 현장 신호설비 상태정보는 ASCS의 RTPS에서 자율분산 프로토콜로 포맷이 변경되어 데이터필드에 전송된다. 라인센터의 LIS는 라인경합관리 서브시스템을 위한 본선 열차 실적 관리 및 라인모니터링 서브시스템을 위한 본선운행열차의 열차번호 표시를 처리한다.

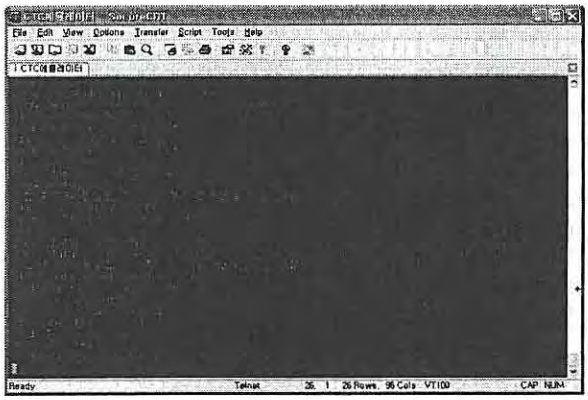


그림 5-51 라인센터 열차추적

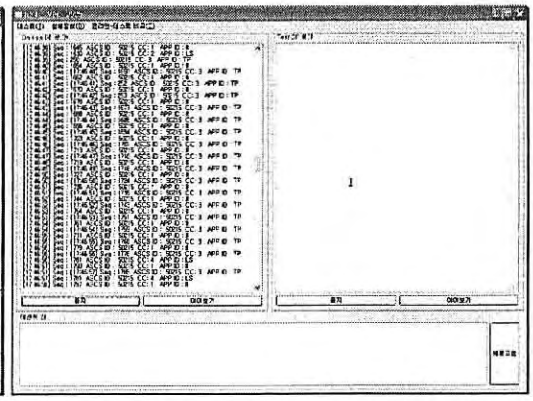


그림 5-52 EXT를 이용한 모니터링

ADTMS의 자율분산관리자는 데이터필드에 송수신되는 메시지를 로그파일로 저장하여 시스템의 데이터 처리 결과를 모니터링 할 수 있도록 한다.

```
18:20:08.444 : 41 53 43 53 01 01 00 02 01 01 00 02 00 00 00 01 00 00 00 00 01 00 00 00 80 00 00 01 01 01 00 88 80 00 03
ff 01 01 00 0a 00 00 c3 c2 00 00 00 81 00 00 00 00 00 2c 47 df 80 00 03 ff 14 22 80 68 00 ca 01 30 30 32 30 32 00 cc 01
30 30 32 30 34 00 c9 01 30 30 32 30 31 00 cb 01 30 30 32 30 33 00 c5 01 30 30 31 39 37 00 c4 01 30 30 31 39 36 00 c8 01
30 30 32 30 30 00 c6 01 30 30 31 39 38 00 c7 01 30 30 31 39 39
18:20:08.507 : 41 53 43 53 01 01 00 02 01 01 00 02 00 00 00 01 00 00 00 00 01 00 00 00 80 00 00 01 00 01 00 98 80 00 03
ff 01 01 00 0a 00 00 c4 27 00 00 00 81 00 00 00 00 00 2c 47 df 80 00 03 ff 14 22 80 68 00 84 01 30 30 31 33 32 00 7a 01
30 30 31 32 32 00 7c 01 30 30 31 32 34 00 7b 01 30 30 31 32 33 00 83 01 30 30 31 33 31 00 85 01 30 30 31 33 33 00 78 01
30 30 31 32 30 00 77 01 30 30 31 31 39 00 75 01 30 30 31 31 37 00 76 01 30 30 31 31 38 00 79 01 30 30 31 32 31
18:20:08.507 : 41 53 43 53 01 01 00 02 01 01 00 02 00 00 00 01 00 00 00 00 01 00 00 00 80 00 00 01 01 01 00 98 80 00 03
ff 01 01 00 0a 00 00 c4 27 00 00 00 81 00 00 00 00 00 2c 47 df 80 00 03 ff 14 22 80 68 00 84 01 30 30 31 33 32 00 7a 01
30 30 31 32 32 00 7c 01 30 30 31 32 34 00 7b 01 30 30 31 32 33 00 83 01 30 30 31 33 31 00 85 01 30 30 31 33 33 00 78 01
30 30 31 32 30 00 77 01 30 30 31 31 39 00 75 01 30 30 31 31 37 00 76 01 30 30 31 31 38 00 79 01 30 30 31 32 31
18:20:09.585 : 41 53 43 53 01 01 00 00 01 01 00 00 49 61 cf 76 00 00 05 2f 00 00 00 00 00 00 00 00 00 01 00 c3 49 61 d0
ec 01 01 00 01 00 00 c4 27 54 4d 53 31 49 49 00 00 00 00 00 00 00 01 3d 4c 49 61 d0 ec 12 8d 01 09 02 01 09 01 01 09 01
09 01 01 01 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 01 00 01 00 01
01 01 01 01 01 01 13 01 13 01 13 01 00 00 01 01 01 01 13 01 01 01 01 01 01 01 01 01 01 01 01 01 01 00 00 00 00 00 00
18:20:10.444 : 41 53 43 53 01 01 00 00 01 01 00 00 49 61 cf 76 00 00 05 30 00 00 00 00 00 00 00 00 00 01 00 c3 49 61 d0
ed 01 01 00 01 00 00 c4 27 54 4d 53 31 49 49 00 00 00 00 00 00 00 01 3d 4c 49 61 d0 ed 12 8d 01 09 02 01 09 01 01 09 01
09 01 01 01 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 01 00 01 00 01
01 01 01 01 01 01 13 01 13 01 13 01 00 00 01 01 01 01 13 01 01 01 01 01 01 01 01 01 01 01 01 01 01 00 00 00 00 00 00
18:20:10.663 : 41 53 43 53 01 01 00 02 01 01 00 02 00 00 00 01 00 00 00 00 01 00 00 00 80 00 00 01 00 01 00 f0 80 00 03
ff 01 01 00 0a 00 00 c3 7c 00 00 00 81 00 00 00 00 00 2c 47 df 80 00 03 ff 14 22 80 68 01 26 01 30 30 32 39 34 01 21 01
30 30 32 38 39 01 22 01 30 30 32 39 30 01 23 01 30 30 32 39 31 01 24 01 30 30 32 39 32 01 25 01 30 30 32 39 33 01 27 01
30 30 32 39 35 01 20 01 30 30 32 38 38 01 28 01 30 30 32 39 36 01 2d 01 30 30 33 30 31 01 2a 01 30 30 32 39 38 01 1e 01
30 30 32 38 36 01 2c 01 30 30 33 30 30 01 39 01 30 30 33 31 33 01 38 01 30 30 33 31 32 01 3a 01 30 30 33 31 34 01 35 01
30 30 33 30 39 01 1f 01 30 30 32 38 37 01 36 01 30 30 33 31 30 01 29 01 30 30 32 39 37 01 2b 01 30 30 32 39 39 01 37 01
30 30 33 31 31
```

```
18:20:10.663 : 41 53 43 53 01 01 00 02 01 01 00 02 00 00 00 01 00 00 00 00 01 00 00 00 80 00 00 01 01 01 00 f0 80 00 03
ff 01 01 00 0a 00 00 c3 7c 00 00 00 81 00 00 00 00 00 2c 47 df 80 00 03 ff 14 22 80 68 01 26 01 30 30 32 39 34 01 21 01
30 30 32 38 39 01 22 01 30 30 32 39 30 01 23 01 30 30 32 39 31 01 24 01 30 30 32 39 32 01 25 01 30 30 32 39 33 01 27 01
30 30 32 39 35 01 20 01 30 30 32 38 38 01 28 01 30 30 32 39 36 01 2d 01 30 30 33 30 31 01 2a 01 30 30 32 39 38 01 1e 01
30 30 32 38 36 01 2c 01 30 30 33 30 30 01 39 01 30 30 33 31 33 01 38 01 30 30 33 31 32 01 3a 01 30 30 33 31 34 01 35 01
30 30 33 30 39 01 1f 01 30 30 32 38 37 01 36 01 30 30 33 31 30 01 29 01 30 30 32 39 37 01 2b 01 30 30 32 39 39 01 37 01
30 30 33 31 31
18:20:10.725 : 41 53 43 53 01 01 00 02 01 01 00 02 00 00 00 01 00 00 00 00 01 00 00 00 80 00 00 01 00 01 00 88 80 00 03
ff 01 01 00 0a 00 00 c3 c2 00 00 00 81 00 00 00 00 00 2c 47 df 80 00 03 ff 14 22 80 68 00 ca 01 30 30 32 30 32 00 cc 01
30 30 32 30 34 00 c9 01 30 30 32 30 31 00 cb 01 30 30 32 30 33 00 c5 01 30 30 31 39 37 00 c4 01 30 30 31 39 36 00 c8 01
30 30 32 30 30 00 c6 01 30 30 31 39 38 00 c7 01 30 30 31 39 39
18:20:10.725 : 41 53 43 53 01 01 00 02 01 01 00 02 00 00 00 01 00 00 00 00 01 00 00 00 80 00 00 01 01 01 00 88 80 00 03
ff 01 01 00 0a 00 00 c3 c2 00 00 00 81 00 00 00 00 00 2c 47 df 80 00 03 ff 14 22 80 68 00 ca 01 30 30 32 30 32 00 cc 01
30 30 32 30 34 00 c9 01 30 30 32 30 31 00 cb 01 30 30 32 30 33 00 c5 01 30 30 31 39 37 00 c4 01 30 30 31 39 36 00 c8 01
30 30 32 30 30 00 c6 01 30 30 31 39 38 00 c7 01 30 30 31 39 39
18:20:10.788 : 41 53 43 53 01 01 00 02 01 01 00 02 00 00 00 01 00 00 00 00 01 00 00 00 80 00 00 01 00 01 00 98 80 00 03
ff 01 01 00 0a 00 00 c4 27 00 00 00 81 00 00 00 00 00 2c 47 df 80 00 03 ff 14 22 80 68 00 84 01 30 30 31 33 32 00 7a 01
30 30 31 32 32 00 7c 01 30 30 31 32 34 00 7b 01 30 30 31 32 33 00 83 01 30 30 31 33 31 00 85 01 30 30 31 33 33 00 78 01
30 30 31 32 30 00 77 01 30 30 31 31 39 00 75 01 30 30 31 31 37 00 76 01 30 30 31 31 38 00 79 01 30 30 31 32 31
18:20:10.788 : 41 53 43 53 01 01 00 02 01 01 00 02 00 00 00 01 00 00 00 00 01 00 00 00 80 00 00 01 01 01 00 98 80 00 03
ff 01 01 00 0a 00 00 c4 27 00 00 00 81 00 00 00 00 00 2c 47 df 80 00 03 ff 14 22 80 68 00 84 01 30 30 31 33 32 00 7a 01
30 30 31 32 32 00 7c 01 30 30 31 32 34 00 7b 01 30 30 31 32 33 00 83 01 30 30 31 33 31 00 85 01 30 30 31 33 33 00 78 01
30 30 31 32 30 00 77 01 30 30 31 31 39 00 75 01 30 30 31 31 37 00 76 01 30 30 31 31 38 00 79 01 30 30 31 32 31
18:20:12.179 : 41 53 43 53 01 01 00 00 01 01 00 00 49 61 cf 76 00 00 05 31 00 00 00 00 00 00 00 00 00 01 00 c3 49 61 d0
ef 01 01 00 01 00 00 c4 27 54 4d 53 31 49 49 00 00 00 00 00 00 00 01 3d 4c 49 61 d0 ef 12 8d 01 09 02 01 09 01 01 09 01
09 01 01 01 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 01 00 01 00 01
01 01 01 01 01 01 13 01 13 01 13 01 00 00 01 01 01 01 13 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 00 00 00 00 00 00
```

그림 5-53 ADTMS 테스트 로그 데이터(부분)

2. 핵심서브시스템 테스트

가. 테스트 환경

(1) RTPS 기능 테스트 환경

- 현장 신호설비 표시정보 생성
- 현장 상태정보 송신

현장 신호설비 표시정보 생성 기능과 현장 상태정보 송신 기능은 테스트 데이터를 발생하는 툴과 결과를 확인하는 툴이 사용된다.

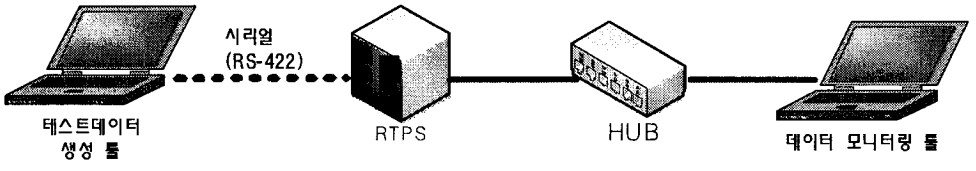


그림 5-54 RTPS ILI-현장 신호설비 표시정보 기능-환경구성

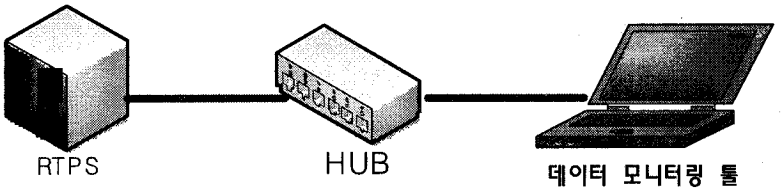


그림 5-55 RTPS ILI-환경구성 기능-환경구성

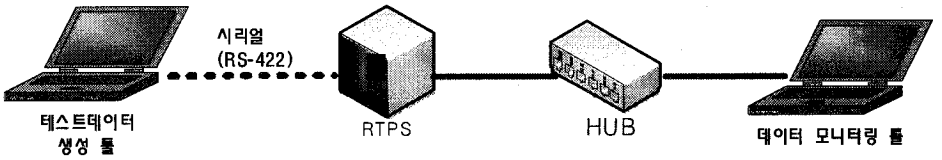


그림 5-56 RTPS ILI-현장 제어메시지 전송 기능-환경구성

열차추적 기능을 위한 시스템 구성은 아래 그림 5-57과 같다.

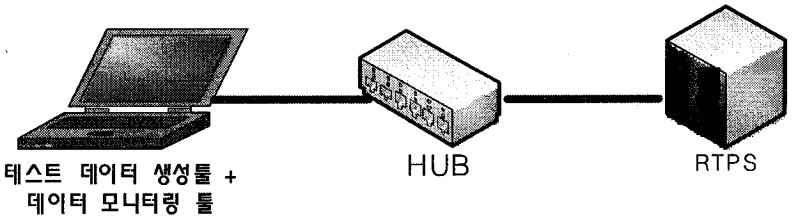


그림 5-57 RTPS 열차추적 기능-환경구성

(2) TMS 기능 테스트 환경

TMS의 기능 테스트를 위한 환경은 그림 5-58과 같다.

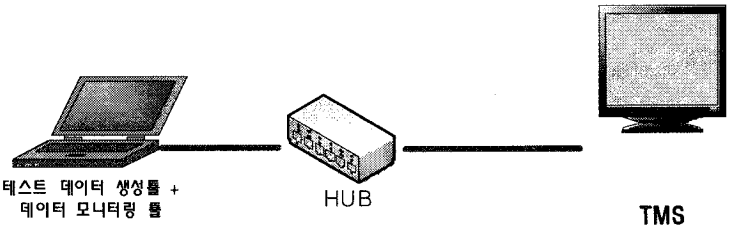


그림 5-58 TMS 기능 테스트 환경

나. 데이터필드와 CC

ASCS 핵심 서브시스템의 데이터필드는 1개로 구성된다. 현재의 시스템 설계 및 구조가 여러 개의 노선을 수용할 수 있고 이를 위해 여러 개의 데이터필드를 구성

할 수도 있으나 시스템의 규모가 작으므로 데이터필드는 1개를 사용하였다.

ASCS에 적용되는 CC는 6가지 종류로 구분하였다. CC 중에서 일부는 실제 시스템으로 적용하는 경우에 사용되지 않고 시스템 종합 테스트 단계에서만 사용된다.

ASCS에 사용된 CC의 종류는 프로토콜 정의 문서에 기록되어 있다.

다. 열차추적 서브시스템(RTPS)의 기능 테스트

실시간 열차위치추적 서브시스템의 프로세스별 세부 기능에 대해서 통합 기능테스트를 수행하기 위한 요구사항 기능은 다음과 같다.

□ 열차번호 관리 프로세스

- 열차 위치 추적
- 역상태 정보 주기적 송신
- 열차번호 대조 기능
- 열차번호 관리)
- 센터 스케줄 수신
- TMS 스케줄 수신
- 착발 실적발생
- 시뮬레이션 기준시각 동기화

□ 표시정보 관리 프로세스

- 현장정보상태 수신
- 현장정보상태 송신
- 현장제어명령 수행

RTPS의 통합 기능 테스트를 구성하기 위해서 역 에뮬레이터와 이벤트 에뮬레이터, CTC 에뮬레이터, 두 개 역 에 대한 RTPS 서브시스템, 한 역에 대한 TMS, RCS로 구성한다.

라. 경로제어 서브시스템(RCS)의 기능 테스트

RCS 서브시스템의 개별 기능에 대해서 통합 기능 테스트를 수행하기 위한 요구사항 기능은 다음과 같다.

□ 경로제어 서브시스템의 요구사항

- 진로 유효성 체크
- 자동 진로 제어
- 센터스케줄 수신
- TMS 스케줄 수신
- 착발 실적 수신
- 시뮬레이션 기준 시각 수신

RCS의 통합 기능 테스트를 구성하기 위해서 현장 LDTS를 모방하는 역 에뮬레이터, 라인 정보관리 서브시스템을 모방하는 CTC에뮬레이터와 한역에 대한 RTPS, RCS, TMS와 인접역에 대한 RTPS로 구성된다. 역 에뮬레이터를 제외한 서브시스템들은 모두 ADM을 통해서 송수신한다. 이외에 연속적인 열차 투입을 위해서 이벤트 에뮬레이터를 사용한다.

마. 열차 모니터링 서브시스템(TMS)의 기능 테스트

열차 모니터링 서브시스템의 프로세스별 세부 기능에 대해서 통합 기능테스트를 수행하기 위한 요구사항 기능은 다음과 같다.

열차 모니터링 서브시스템의 기능 테스트에 대한 구체적인 절차와 결과 그리고 검증 포인트에 대해서 설명하도록 한다. 열차 모니터링 서브시스템의 특성상(GUI 가존재함) 테스트 결과를 보여주기 위해 화면을 캡처하여 표현하였고 TMS 서브시스템의 통합 기능테스트를 구성하기 위해서 역 에뮬레이터, CTC 에뮬레이터, RTPS 서브시스템, RCS 서브시스템으로 구성한다.

□ 화면표시 프로세스

- 선형 정보 표시
- 열차 상태 표시
- 이벤트 및 알람표시

□ 현장제어 프로세스

- 운영모드 제어
- 스케줄 정보관리
- 선로전환기 제어
- 진로 제어
- 열차번호 제어
- 신호기 제어

바. 홈 리스케줄링 연동시험

(1) 홈 리스케줄링 자체 기능 테스트

(가) 홈 리스케줄링 자체 기능 테스트 개요

홈 리스케줄링 자체 기능 테스트는 홈 리스케줄링 프로세스와 ASCS 1SET내의 타 서브시스템(RTPS, RCS, SMS) 및 프로세스간의 연동 시험, 인접역 및 라인경합관리와의 연동이 되지 않는 상태에서 수행할 수 있는 홈 리스케줄링 경합 검지 및 해소 기능 테스트를 목적으로 한다.

(나) 테스트 환경 구성

① 시험환경 구성

홈 리스케줄링 자체기능 테스트의 환경구성은 그림5-59과 같이 청량리역 ASCS 1SET와 역 에뮬레이터, 이벤트 에뮬레이터로 구성 하였다.

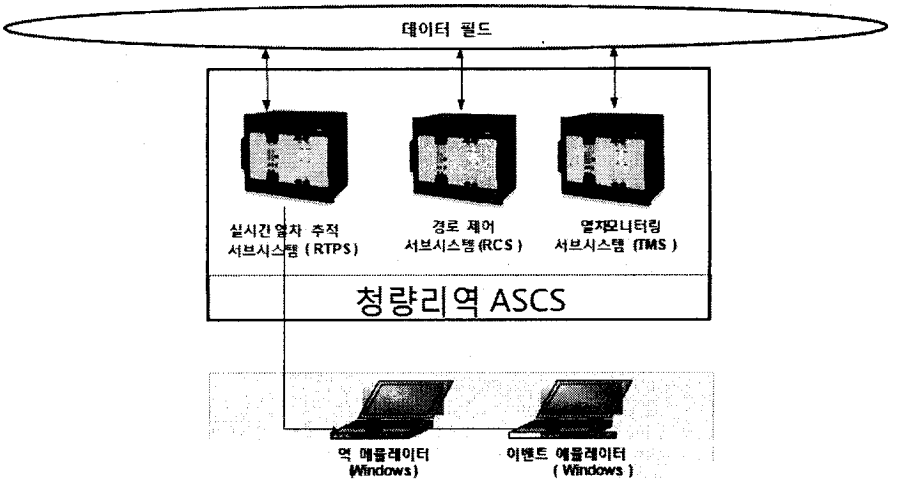


그림 5-59 자체테스트 시험환경 구성

② 테스트 스케줄 구성

시험에 사용되는 청량리역 스케줄은 총 14개의 열차로 구성되어 있으며, 이중 4개 열차는 청량리역에서 시발 혹은 종착하는 열차이며 나머지 10개 열차(성북방면 하행 열차 3대, 중랑방면 하행열차 4대, 왕십리방면 상행열차 3대)는 청량리역에서 정차 후 통과하는 열차이다.

표 5-9 자체기능시험 테스트 스케줄

열차번호	순서	계획 도착시간	계획도착진로	계획 출발시간	계획출발진로	정차 시간
A0800	0	080400	130126RT2B_11T			
K0801	0	080700	130126RT2B_13T	080730	130126RT3A_B155T	000030
K0802	0	080800	130126RT1A_9T	080830	130126RT4H_B5002T	000030
K0803	0	081030	130126RT2C_93AT			
A0806	0			081200	130126RT4D_B5002T	
K0805	0	081500	130126RT2B_13T	081530	130126RT3A_B155T	000030
K0807	0			081400	130126RT5A_B158AT	
K0804	0	081600	130126RT1A_9T	081630	130126RT4H_B5002T	000030
A0809	0			081800	130126RT4B_B158AT	
A0808	0	082200	130126RT2A_4T			
K0810	0	082400	130126RT1A_9T	082430	130126RT4H_B5002T	000030
K0811	0	082700	130126RT2B_13T	082730	130126RT3A_B155T	000030
K0813	0			082600	130126RT5A_B158AT	
K0812	0	082830	130126RT2C_93AT			

(다) 홈 리스케줄링 자체 기능 테스트 시나리오 설계

① HR 경합 검지 테스트

- 고장발생으로 인한 진로 사용 불가 검지

궤도, 신호기, 선로전환기 등의 연동진로를 구성하는 Elements에 고장이 발생되어 고장정보가 RTPS를 통해 현장에서 올라오면 HR에서는 이를 확인한 후 고장이 발생한 Elements가 영향을 주는 모든 진로의 사용가능여부를 사용불능으로 설정하며 사용불능으로 설정된 진로를 사용 예정된 열차스케줄이 있다면 열차번호, 도착/출발 진로여부, 진로명을 사용자에게 알람 하여 준다.

- 궤도 점유순서 불일치 검지

역 운영자의 열차 출발 순서 변경 및 결합스케줄 변경으로 인하여 진로상의 열차가 점유하는 최종의 궤도의 도착순서와 출발순서를 비교하여 불일치한 열차스케줄이 있다면 경합이 발생된 열차번호들과 발생 최종 궤도를 사용자에게 알람 하여 준다. 예를 들어, A열차가 9번 플랫폼으로 8시에 도착하여 8시 30분에 출발하는 스케줄이고 B열차가 동일한 9 플랫폼으로 8시 15분에 도착하여 8시 20분에 출발하는 스케줄이라면 물리적으로 동일한 9번 플랫폼을 점유하는 열차의 출 도착순서가 일치하지 않게 되어 A열차, B열차, 9T궤도 점유 이상이라는 알람이 생성된다.

② HR 경합 해소 테스트

- 대안진로 입력으로 진로 사용 불가 해소

궤도, 신호기, 선로전환기 등의 연동진로를 구성하는 Elements에 고장이 발생되어 진로사용불가에 따라 해당 진로를 사용하는 열차스케줄의 열차번호, 도착/출발 진로여부, 진로명 정보가 알람 되었을 때, 사용자가 이를 우회할 수 있는 여러 대안 진로들을 입력하여 주면 홈 리스케줄링에서는 여러 대안 진로 중 HR 대상열차들의 지연시간을 최소화하는 최적의 진로를 선택하여 변경 스케줄을 생성한다.

- 궤도 점유순서 불일치 해소

최종궤도에서의 열차 출 도착순서가 일치하지 않는 열차들에 대한 열차번호 및 궤도 정보가 알람이 되었을 때, 사용자가 최종궤도의 점유순서를 일치 하도록 예측

스케줄을 조정 요구하여 주면 홈 리스케줄링에서는 조정 요구된 스케줄을 바탕으로 HR 대상열차들의 지연시간을 최소화하는 최적의 시간을 선택하여 변경 스케줄을 생성한다.

(라) 자체 기능 테스트 결과

앞에서 설계된 테스트 시나리오를 바탕으로 타 서브시스템(RTPS, RCS, SMS) 및 프로세스간의 연동 시험, 인접역 및 라인경합관리와의 연동이 되지 않는 상태에서 수행할 수 있는 홈 리스케줄링 경합 검지 및 해소 기능 테스트 항목에 대하여 테스트하는 절차와 시험조건, 시험 결과에 대해서 기술한다.

① 궤도고장 경합검지 기능

그림 5-60과 같이 사용예정인 진로상의 궤도에서 고장이 발생하면 관련 경로의 사용가능여부속성이 불가로 변경하고 해당진로가 기본스케줄에 포함되어있다면 경합검지 DLL에서 경합이 발생된 열차번호, 순서, 도착/출발, 진로명을 알람 한다.

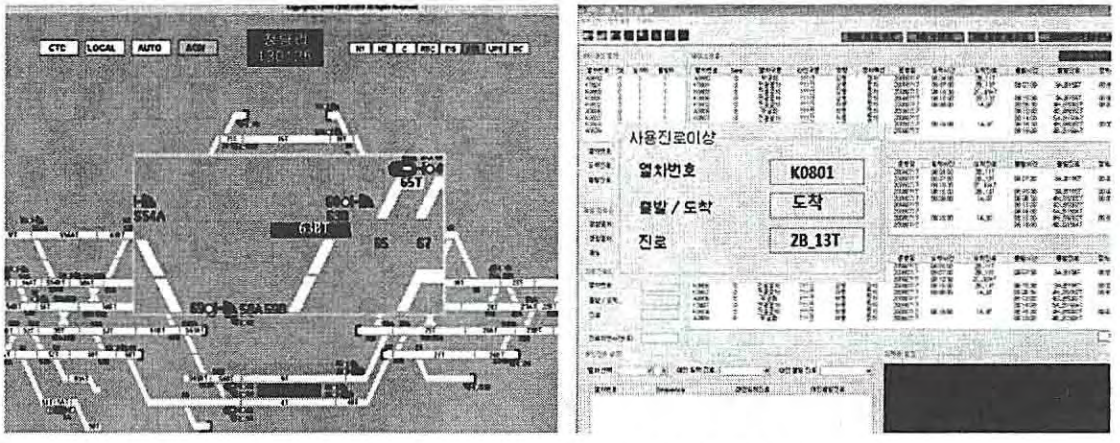


그림 5-60 궤도고장 경합검지 기능

- 시험조건

테스트스케줄1, HR대상열차수 10, HR대상시간 10분, 삭제여유시각 60초, 경보여유시각 60초, 수정불가시각 1분, 라인별 출발순서 고정

- 시험절차 및 결과

표 5-10 궤도고장 경합검지 시험절차 및 결과

번호	입력 및 결과
1	입력 - 청량리역 에뮬레이터를 통해 궤도고장 입력 궤도고장 : 63BT 고장(@ 080600)
	결과 - 궤도고장으로 인해 tv_route의 궤도사용가능 정보가 사용불능으로 변경되므로 경합검지가 실행되며 해당 궤도 불능으로 영향을 받는 진로를 사용 중인 스케줄이 있다면 알람을 발생시킨다. 궤도고장 : 알람 발생(K0801 | 도착 | 2B_13T)
2	입력 - 청량리역 에뮬레이터를 통해 궤도정상 입력 궤도고장 : 63BT 정상(@ 080700)
	결과 - 궤도정상으로 tv_route의 궤도사용가능 정보가 사용가능으로 변경되므로 경합검지가 실행되며 해당 궤도 상태가 정상으로 바뀜에 따라 진로 역시 사용가능으로 변경되고 경합 알람은 사라진다. 궤도고장 : 경합알람 사라짐.

② 선로전환기고장 경합검지 기능

그림 5-61과 같이 사용예정인 진로상의 선로전환기에서 고장이 발생하면 관련 경로의 사용가능여부속성이 불가로 변경하고 해당진로가 기본스케줄에 포함되어있다면 경합검지 DLL에서 경합이 발생된 된 열차번호, 순서, 도착/출발, 진로명을 알람 한다.

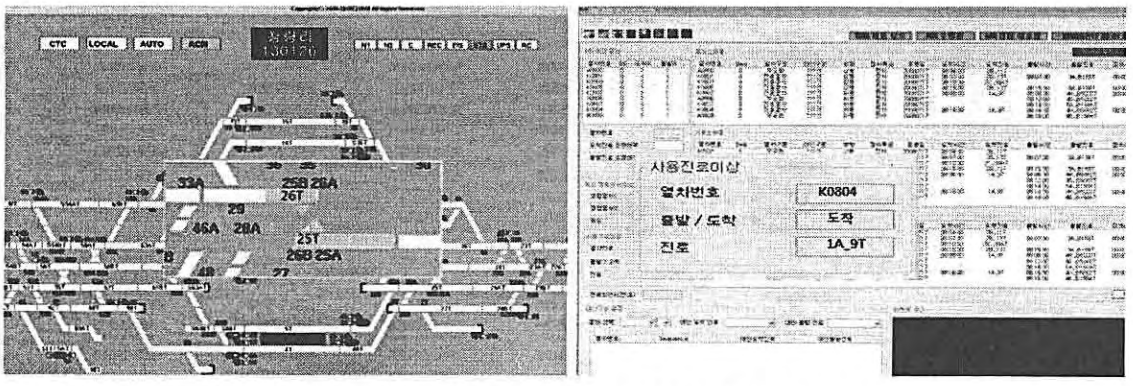


그림 5-61 선로전환기고장 경합검지

- 시험조건

테스트스케줄1, HR대상열차수 10, HR대상시간 10분, 삭제여유시각 60초, 경보여

유시각 60초, 수정불가시각 1분, 라인별 출발순서고정

- 시험절차 및 결과

표 5-11 선로전환기고장 경합검지 시험절차 및 결과

번호	입력 및 결과
1	입력 - 청량리역 에뮬레이터를 통해 선로전환기 고장 입력 선로전환기고장 : 25A 고장(@ 081300)
	결과 - 선로전환기 고장으로 인해 tv_route의 궤도사용가능 정보가 사용불능으로 변경되므로 경합검지가 실행되며 해당 궤도 불능으로 영향을 받는 진로를 사용 중인 스케줄이 있다면 알람을 발생시킨다. 선로전환기고장 : 알람 발생(K0804 | 도착 | 1A_9T)
2	입력 - 청량리역 에뮬레이터를 통해 선로전환기 정상입력 선로전환기고장 : 25A 정상(@081500)
	결과 - 선로전환기 정상으로 tv_route의 궤도사용가능 정보가 사용가능으로 변경되므로 경합검지가 실행되며 해당 궤도 상태가 정상으로 바뀜에 따라 진로 역시 사용가능으로 변경되고 경합 알람은 사라진다. 선로전환기고장 : 경합알람 사라짐.

③ 신호기고장 경합검지 기능

아래 그림5-62와 같이 사용예정인 진로상 신호기에 고장이 발생하면 관련 경로의 사용가능여부속성이 불가로 변경하고 해당진로가 기본스케줄에 포함되어있다면 경합검지 DLL에서 경합이 발생된 된 열차번호, 순서, 도착/출발, 진로명을 알람 한다.

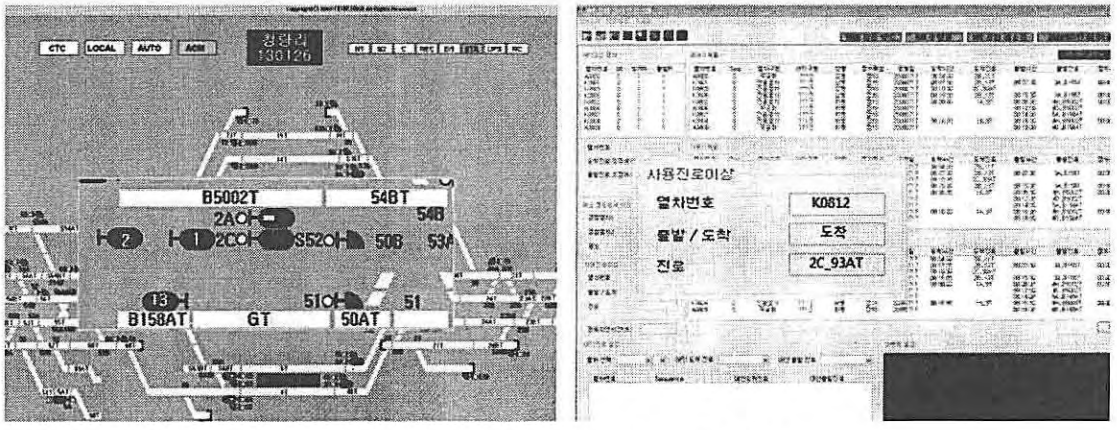


그림 5-62 신호기고장 경합검지

- 시험조건

테스트스케줄1, HR대상열차수 10, HR대상시간 10분, 삭제여유시각 60초, 경보여유시각 60초, 수정불가시각 1분, 라인별 출발순서고정.

- 시험절차 및 결과

표 5-12 신호기고장 경합검지 시험절차 및 결과

번호	입력 및 결과
1	입력 - 청량리역 에뮬레이터를 통해 신호기고장 입력 신호기고장 : 2C 고장(@ 082500)
	결과 - 신호기 고장으로 인해 tv_route의 궤도사용가능 정보가 사용불능으로 변경되므로 경합검지가 실행되며 해당 궤도 불능으로 영향을 받는 진로를 사용 중인 스케줄이 있다면 알람을 발생시킨다. 신호기고장 : 알람 발생(K0812 | 도착 | 2C_93AT)
2	입력 - 청량리역 에뮬레이터를 통해 신호기정상 입력 신호기고장 : 2C 정상(@082600)
	결과 - 신호기 정상으로 인해 tv_route의 궤도사용가능 정보가 사용가능으로 변경되므로 경합검지가 실행되며 해당 궤도 상태가 정상으로 바뀜에 따라 진로 역시 사용가능으로 변경되고 경합 알람은 사라진다. 신호기고장 : 경합알람 사라짐.

④ 궤도 점유 순서 이상검지

아래 그림5-63과 같이 역 운영자의 열차 출발 순서 변경 및 결합스케줄 변경으로 인하여 진로상의 열차가 점유하는 최종의 궤도의 도착순서와 출발순서를 비교하여 불일치한 열차스케줄이 있다면 경합이 발생된 열차번호들과 발생 최종 궤도를 사용자에게 알람 하여 준다.

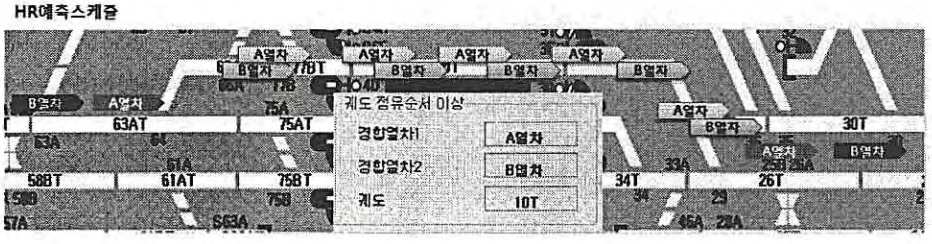


그림 5-63 궤도 점유 순서 이상검지

- 시험조건

테스트스케줄1, HR대상열차수 10, HR대상시간 20분, 삭제여유시각 60초, 경보여유시각 60초, 수정불가시각 1분, 라인별 출발순서고정, 전체지연한도 600초

- 시험절차 및 결과

표 5-13 궤도 점유 순서 이상검지 시험절차 및 결과

번호	입력 및 결과
1	입력 : 예측스케줄 수정 MMI에서 K0802열차의 출발예측시간을 081700이후로 세팅한 후에 "수정"버튼을 누른다.(@080800)
	결과 : K0802열차의 변경출발예측시간이 081700이후로 세팅된다.
2	입력 : "적용"버튼을 눌러 변경된 예측스케줄을 변경 저장 한다.
	결과 : 예측스케줄 변경으로 인해 경합검지가 실행되고 알람발생(K0802 ㅣ K0804 ㅣ 9T).

⑤ 궤도고장 경합해소 기능

궤도고장이 발생하면 관련 경로의 Route-Avail속성이 불가로 변경하고 해당진로가 기본스케줄에 포함되어있다면 경합검지 DLL에서 문제가 된 열차번호, 순서, 도착/출발, 진로명을 알람 하며 문제발생 열차에 대안진로를 입력하여 경합을 해소한다.

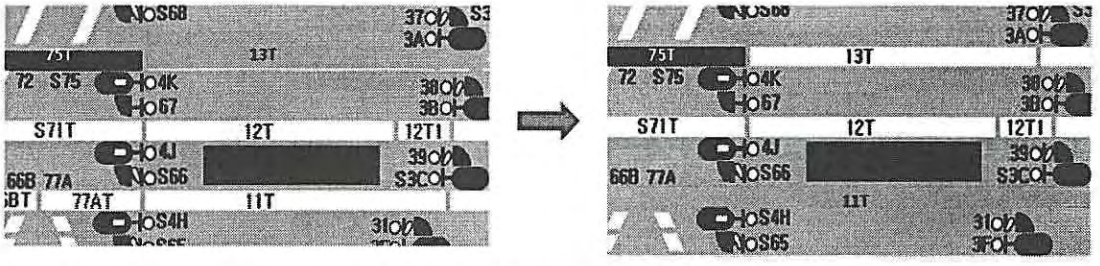


그림 5-64 궤도고장 경합해소 기능

- 시험조건

테스트스케줄1, HR대상열차수 10, HR대상시간 10분, 삭제여유시각 60초, 경보여유시각 60초, 수정불가시각 1분, 라인별 출발순서고정

- 시험절차 및 결과

표 5-14 궤도고장 경합해소 시험절차 및 결과

번호	입력 및 결과
1	입력 : 청량리역 에뮬레이터를 통해 궤도고장 입력 궤도고장 : 75T 고장(@ 082300)
	결과 : 궤도 고장으로 인해 tv_route의 avail 정보가 달라지므로 경합검지 실행 궤도고장 : 알람 발생(K0811 | 도착 | 2B_13T)
2	입력 : HR 대안진로를 입력 K0811 대안진로 (도착 : 2B_11T, 출발: S3C_B155T)입력 후 "추가"버튼 입력
	결과 : K0811 열차의 대안진로(도착 : 2B_11T, 출발: S3C_B155T)가 추가됨.
3	입력 : HR 대안진로를 적용 "적용" 버튼 입력 후 "사용자경합해소" 버튼 입력
	결과 : 경합해소스케줄 생성 및 조회 화면이 뜨며 K0811 열차의 스케줄을 도착 : 2B_11T, 출발: S3C_B155T로 변경 제안
4	입력 : HR 경합해소스케줄 승인 "승인" 버튼 입력
	결과 : K0811 열차의 기본, 제어, 예측 스케줄상에 진로가 도착 : 2B_11T, 출발: S3C_B155T로 변경되며 기본스케줄 변경으로 인해 경합이 발생하지 않음.

⑥ 선로전환기고장 경합해소 기능

- 시험설명

선로전환기 고장이 발생하면 관련 경로의 Route-Avail속성이 불가로 변경하고 해당진로가 기본스케줄에 포함되어있다면 경합검지 DLL에서 문제가 된 열차번호, 순서, 도착/출발, 진로명을 알람 하며 문제발생 열차에 대안진로를 입력하여 경합을 해소 한다.

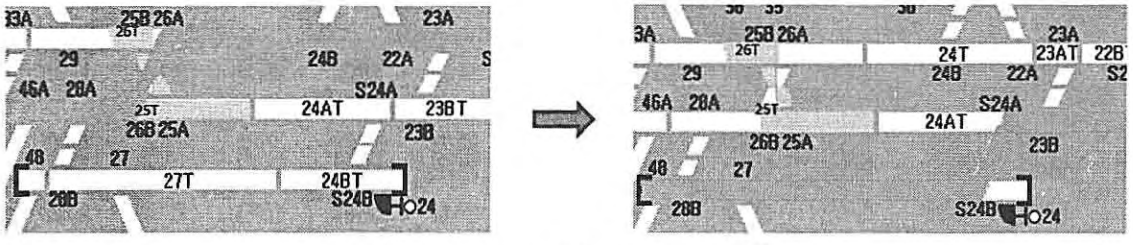


그림 5-65 선로전환기 고장 경합해소

- 시험조건

테스트스케줄1, HR대상 열차수 10, HR대상시간 10분, 삭제여유시각 60초, 경보여유시각 60초, 수정불가시각 1분, 라인별 출발순서 고정

- 시험절차 및 결과

표 5-15 선로전환기 고장 경합해소 시험절차 및 결과

번호	입력 및 결과
1	입력 : 청량리역 에뮬레이터를 통해 선로전환기 고장 입력 선로전환기고장 : 23A 고장(@ 082300)
	결과 : 선로전환기 고장으로 인해 tv_route의 avail 정보가 달라지므로 경합검지 실행 선로전환기고장 : 알람 발생(K0810 | 도착 | 1A_9T)
2	입력 : HR 대안진로를 입력 선로전환기고장 : K0810 대안진로 (도착 : 1A_4T, 출발: 4A_B5002T)입력 후 "추가"버튼 입력
	결과 : K0810 열차의 대안진로(도착 : 1A_4T, 출발: 4A_B5002T)추가됨.
3	입력 : HR 대안진로를 적용 "적용" 버튼 입력 후 "사용자경합해소" 버튼 입력
	결과 : 경합해소스케줄 생성 및 조회 화면이 뜨며 K0815 열차의 스케줄을 도착 : 1A_4T, 출발: 4A_B5002T로 변경 제안
4	입력 : HR 경합해소스케줄 승인 "승인" 버튼 입력
	결과 : K0810 열차의 기본, 제어, 예측 스케줄상에 진로가 도착 : 1A_4T, 출발: 4A_B5002T로 변경되며 기본스케줄 변경으로 인해 경합검지가 다시 실행되어 K0811 | 출발 | 3B_B155AT 경합이 발생 알람.
5	입력 : 청량리역 에뮬레이터를 통해 선로전환기 정상 입력 선로전환기고장 : 23A 정상
	결과 : 경합 알람 사라짐. 선로전환기고장 : 경합알람 사라짐.

⑦ 궤도 점유 순서 이상 해소

역 운영자의 열차 출발 순서 변경 및 결합스케줄 변경으로 인하여 진로상의 열차가 점유하는 최종의 궤도의 도착순서와 출발순서가 불일치하여 경합이 발생된 열차번호와 발생최종궤도 정보가 알람 되어 사용자가 수동으로 예측스케줄을 변경하

여주면 이 정보를 바탕으로 궤도 점유 순서를 지키면서 지연시간을 최소화하도록 변경스케줄을 생성한다.

- 시험조건

테스트스케줄1, HR대상열차수 10, HR대상시간 20분, 삭제여유시각 60초, 경보여유시각 60초, 수정불가시각 1분, 라인별 출발순서고정, 전체지연한도 600초

- 시험절차 및 결과

표 5-16 궤도고장 경합해소 시험절차 및 결과

번호	입력 및 결과
1	입력 : 예측스케줄 수정 MMI에서 K0802열차의 출발예측시간을 081700이후로 세팅한 후에 "수정"버튼을 누른다.(@080800)
	결과 : K0802열차의 변경출발예측시간이 081700이후로 세팅된다.
2	입력: "적용"버튼을 눌러 변경된 예측스케줄을 변경 저장 한다.
	결과 : 예측스케줄 변경으로 인해 경합검지가 실행되고 알람발생(K0802 | K0804 | 9T).
3	입력 : K0802열차의 출발예측시간을 081700이후로 세팅한 후에 "일괄수정"버튼을 누른다.
	결과 : K0802, K0804, K0810, K0815 열차의 변경출발예측시간이 081700이후로 세팅된다.
4	입력 : "적용"버튼을 눌러 변경된 예측스케줄을 변경 저장 한다.
	결과 : K0802, K0804, K0810, K0815 열차의 변경출발예측시간이 081700이후로 변경된다.
5	입력 : "적용"버튼을 눌러 변경된 예측스케줄을 변경 저장 한다.
	결과 : 예측스케줄 변경으로 경합검지가 실행되지만 최종궤도의 도착순서 및 출발순서가 일치하므로 궤도 점유 순서 이상검지는 생기지 않는다.

(2) 홈 리스케줄링 인접역 연동 테스트

(가) 홈 리스케줄링 인접역 연동 기능 테스트 개요

홈 리스케줄링 인접역 연동 기능 테스트는 홈 리스케줄링 프로세스와 인접역

ASCS 간에 연동이 되는 상태에서 수행할 수 있는 홈 리스케줄링 경합 검지 및 해소 기능 테스트를 목적으로 한다.

(나) 테스트 환경 구성

① 시험환경 구성

홈 리스케줄링 인접역 연동기능 테스트의 환경구성은 그림5-66과 같이 청량리역 ASCS 1SET, 인접 3개역(성북, 중랑, 왕십리)의 ASCS, 역 에뮬레이터, 이벤트 에뮬레이터로 구성 하였다.

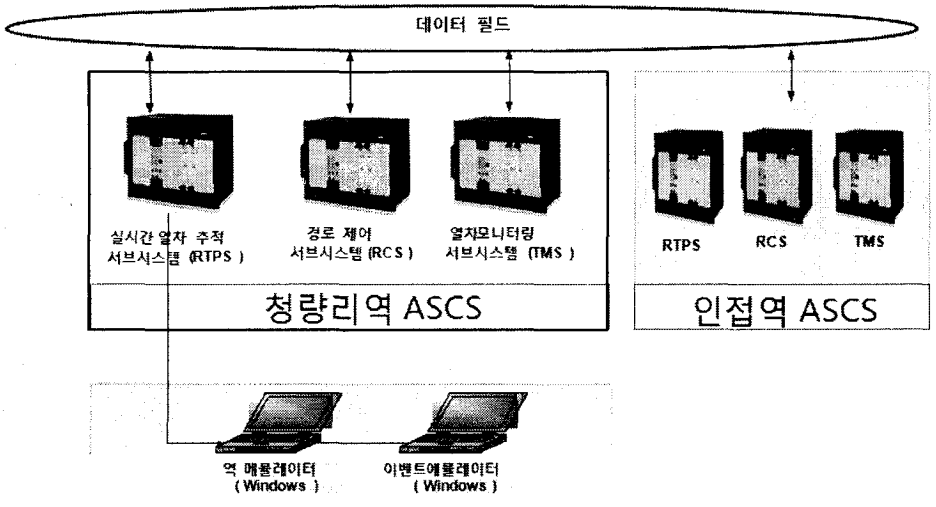


그림 5-66 인접역 연동 테스트 시험환경 구성

② 테스트 스케줄 구성

시험에 사용되는 청량리역 스케줄은 자체기능테스트의 테스트스케줄과 동일하며 이와 연동되어있는 인접역의 테스트스케줄은 아래와 같다.

- 성북

시험에 사용되는 성북 스케줄은 6대의 열차로 청량리에서 하행하는 열차 3대 청량리로 상행하는 열차가 3대이다.

표 5-17 성북 ASCS 본선스케줄

열차 번호	순서	계획 도착시간	계획도착진로	계획 출발시간	계획출발진로	정차 시간
K0803	0			080600	130182RT3D_B129T	
A0808	0			081600	130182RT3G_B129T	
K0807	0	081830	130182RT1A_2T	081900	130182RT4B_B802T	000030
K0812	0			082400	130182RT3D_B129T	
A0809	0	082400	130182RT1A_3T			
K0813	0	083030	130182RT1A_2T	083100	130182RT4B_B802T	000030

- 중랑

시험에 사용되는 중랑 스케줄은 8대의 열차로 청량리에서 하행하는 열차 4대 청량리로 상행하는 열차가 4대이다.

표 5-18 중랑 ASCS 본선스케줄

열차 번호	순서	계획 도착시간	계획도착진로	계획 출발시간	계획출발진로	정차 시간
A0800	0			080000	020032RT3B_B5005T	
K0801	0			080200	020032RT3A_B5005T	
K0805	0			081000	020032RT3A_B5005T	
K0802	0	081230	020032RT1A_4T	081300	020032RT4A_B5010T	000030
A0806	0	081600	020032RT1A_3T			
K0804	0	082000	020032RT1A_4T	082030	020032RT4A_B5010T	000030
K0811	0			082200	020032RT3A_B5005T	
K0810	0	082820	020032RT1A_4T	082850	020032RT4A_B5010T	000030

- 왕십리

시험에 사용되는 왕십리 스케줄은 6대의 열차로 청량리에서 하행하는 열차 3대 청량리로 상행하는 열차가 3대이다.

표 5-19 왕십리 ASCS 본선스케줄

열차 번호	순서	계획 도착시간	계획도착진로	계획 출발시간	계획출발진로	정차 시간
K0802	0			080400	130104RT4B_51T1T	
K0804	0			081200	130104RT4B_51T1T	
K0801	0	081100	130104RT2A_6T	081130	130104RT3B_B157T	000030

K0805	0	081930	130104RT2A_6T	082000	130104RT3B_B157T	000030
K0810	0			082000	130104RT4B_51T1T	
K0811	0	083130	130104RT2A_6T	083200	130104RT3B_B157T	000030

(다) 홈 리스케줄링 인접역 연동 테스트 시나리오 설계

① HR 경합 검지 테스트

- 인접역 지연 발생으로 인한 경합검지

인접역에서 출발지연이 발생된 경우 해당 역 HR에서는 인접역 RTPS로부터 생성된 열차 지연출발 정보를 확인한 후 지연발생 열차의 해당역 도착 시간을 재 예측한다. 재 예측된 예측스케줄과 기본스케줄과의 전체 지연발생 시간이 사용자가 지정한 한도시간을 초과할 경우 지연을 검지하여 전체지연발생시간을 알람 한다.

- 도착순서불일치 경합검지

열차 예측 도착시간 변경으로 인하여 기본스케줄상의 상하행 라인별 열차 도착순서와 예측스케줄상의 열차 도착순서가 일치 하지 않을 때 경합이 발생된 열차의 열차번호, 상하행정보, 라인정보를 사용자에게 알람 하여 준다.

② HR 경합 해소 테스트

- 인접역 지연 발생으로 인한 경합해소

인접역에서 출발지연으로 발생된 지연이 사용자가 지정한 한도시간을 초과하여 알람 되었을 때, 전체지연시간을 최소화 하는 변경 스케줄을 생성한다.

- 도착순서불일치 경합해소

기본스케줄상의 상하행 라인별 열차 도착순서와 예측스케줄상의 열차 도착순서가 일치 하지 않는 열차들에 대한 열차의 열차번호, 상하행정보, 라인정보가 알람이 되었을 때, 사용자가 수동으로 예측스케줄을 조정하지 않으면 기본스케줄상의 도착순서에 맞추어서 예측스케줄상의 열차 도착 및 출발 시간을 최적으로 조정한다.

(라) 인접역 연동 기능 테스트 결과

이 장에서는 앞에서 설계된 테스트 시나리오를 바탕으로 홈 리스케줄링 프로세스와 인접역 ASCS 간에 연동이 되는 상태에서 수행할 수 있는 홈 리스케줄링 경합 검지 및 해소 기능 테스트 항목에 대하여 테스트하는 절차와 시험조건, 시험 결과에 대해서 기술한다.

① 인접역 지연 발생으로 인한 경합검지

인접역 출발지연이 발생하면 인접역에서 출발실적생성 시 해당역에서 예측스케줄이 수정변경이 발생된 후 기본스케줄상의 도착순서와 예측스케줄상의 도착순서가 일치하지 않을 때에는 경합검지DLL에서 열차번호, 순서, 상/하행, 라인번호를 알람한다.

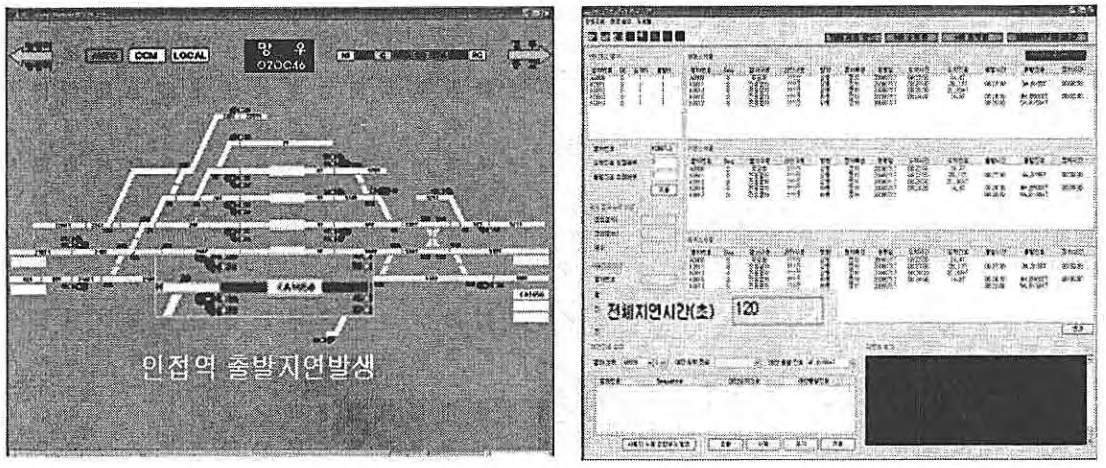


그림 5-67 인접역 지연발생으로 인한 경합검지

- 시험조건

테스트스케줄1, HR대상열차수 10, HR대상시간 10분, 삭제여유시각 60초, 경보여유시각 60초, 수정불가시각 1분, 라인별 출발순서고정, 전체지연시간한도 30초

- 시험절차 및 결과

표 5-20 인접역 지연발생 경합 시험절차 및 결과

번호	입력 및 결과	비고
1	입력 : MTMS에서 중랑, 성북, 왕십리 열차 지연 출발 중랑지연: 열차A0800을 30초 지연시켜 080130에 중랑역에서 출발시킨다.	
	결과 : 중랑 지연: 열차A0800의 예측스케줄상의 도착시간이 080500을 초과한다.	
2	입력 : 예측시간 변경으로 인한 Trigger가 발생하여 시스템에서 자동으로 경합검지를 실행한다.	
	결과 : 전체지연한도 60초를 초과하므로 경합이 검지되고 자동으로 경합해소가 실행된다.	경합해소전 지연시간을 체크한다.

② 도착순서불일치 경합검지

열차 예측 도착시간 변경으로 인하여 기본스케줄상의 도착순서와 예측스케줄상의 도착순서가 일치하지 않을 때에는 경합검지DLL에서 열차번호, 순서, 상/하행, 라인번호를 알람 한다.

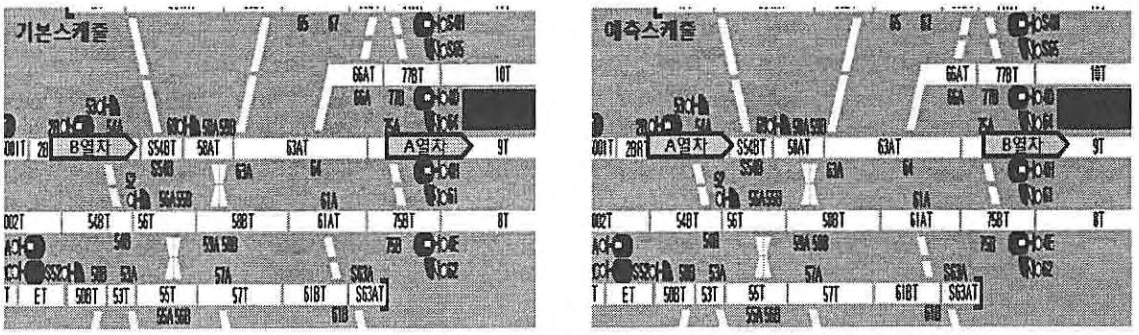


그림 5-68 도착순서불일치 경합검지

- 시험조건

테스트스케줄1, HR대상열차수 10, HR대상시간 10분, 삭제여유시각 60초, 경보여유시각 60초, 수정불가시각 1분, 라인별 출발순서고정, 전체지연시간한도 360초

- 시험절차 및 결과

표 5-21 도착순서불일치 경합검지 시험절차 및 결과

번호	입력 및 결과	비고
1	입력 : 다중열차운행시뮬레이터에서 중랑, 성북, 왕십리 열차 지연 출발 중랑 지연: 열차 A0800을 1분 지연시켜 080130에 중랑역에서 출발시킨다.	
	결과 : 중랑 지연 - 열차A0800의 예측스케줄상의 도착시간이 080530을 초과한다. 도착순서 불일치가 발생한다. 로그알람발생(K0801, 상행, 중앙선)	

③ 인접역 지연 발생 경합해소

인접역 지연 발생으로 인하여 지연시간 한도 초과가 발생하면 자동으로 경합해소 알고리즘이 실행되어 계획된 스케줄의 진로를 그대로 유지할 때, HR 대상열차들의 도착시간 및 출발시간을 조정하여 전체 지연시간을 최소화 한다.

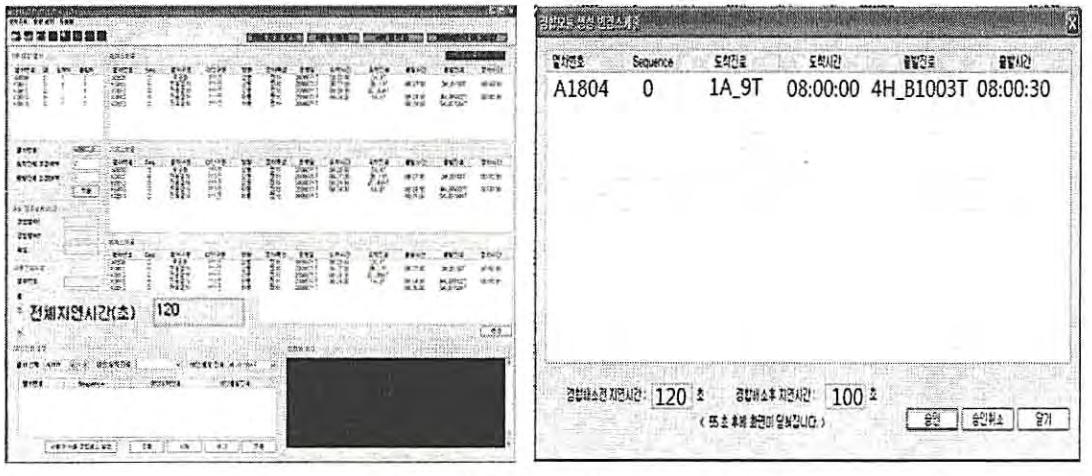


그림 5-69 인접역 지연 발생 경합해소

- 시험조건

테스트스케줄1, HR대상열차수 10, HR대상시간 10분, 삭제여유시각 60초, 경보여유시각 60초, 수정불가시각 1분, 라인별 출발순서고정, 전체지연시간한도 60초

- 시험절차 및 결과

표 5-22 인접역 지연 발생 경합해소 시험절차 및 결과

번호	입력 및 결과	비고
1	입력 : MTMS에서 중랑, 성북, 왕십리 열차 지연 출발 중랑 지연: 열차A0800을 60초 지연시켜 080130에 중랑역에서 출발 시킨다.	
	결과 : 중랑 지연-열차A0800의 예측스케줄상의 도착시간이 080530을 초과한다.	
2	입력 : 예측시간 변경으로 인한 Trigger가 발생하여 시스템에서 자동으로 경합검지를 실행한다.	
	결과 : 전체지연한도 60초를 초과하므로 경합이 검지되고 자동으로 경합해소가 실행된다. 경합해소전 지연시간을 체크 한다.	
3	입력 : HR 변경스케줄이 생성이 되고 "승인"버튼을 누른다.	
	결과 : 승인된 내용이 제어스케줄로 반영이 되었는지 확인한다. 경합해소후 지연시간을 체크한다.	경합전·후 지연시간 비교.

④ 도착순서불일치 경합검지

열차 예측 도착시간 변경으로 인하여 기본스케줄상의 도착순서와 예측스케줄상의 도착순서가 일치하지 않을 때 사용자의 수동개입이 없을 때에는 자동으로 예측스케줄 도착시간을 수정하여 기본스케줄상의 도착순서와 예측스케줄상의 도착 순서를 일치시켜 경합발생을 해소시킨다.

- 시험조건

테스트스케줄1, HR대상열차수 10, HR대상시간 10분, 삭제여유시각 60초, 경보여유시각 60초, 수정불가시각 1분, 라인별출발순서고정, 전체지연시간한도 360초

- 시험절차 및 결과

표 5-23 도착순서불일치 경합검지 시험절차 및 결과

번호	입력 및 결과
1	입력 : 다중열차운행시뮬레이터에서 중랑, 성북, 왕십리 열차 지연 출발 중랑 지연: 열차 A0800을 1분 지연시켜 080130에 중랑역에서 출발시킨다.
	결과 : 중랑 지연 - 열차A0800의 예측스케줄상의 도착시간이 080530을 초과한다. 도착순서 불일치가 발생한다. 로그알람발생(K0801, 상행, 중앙선)
3	입력: 자동으로 같은 라인의 이후 도착열차(K0801)의 도착시간이 080530이후로 변경된다.
	결과: K0802, K0804, K0810, K0815 열차의 변경출발예측시간이 081700이후로 세팅된다.
4	입력: "적용"버튼을 눌러 변경된 예측스케줄을 변경 저장 한다.
	결과 : 예측스케줄 변경으로 경합검지가 Trigger되지만 기본스케줄의 도착순서와 예측스케줄의 도착순서가 일치하므로 경합검지로 인한 알람은 발생하지 않는다.

(3) 홈 리스케줄링 라인경합관리 연동테스트

(가) 라인경합관리 연동 테스트 개요

홈 리스케줄링 라인경합관리 연동 테스트는 자율분산제어체계에서 계층적 구조로 설계되어있는 경합검지 및 해소기능을 하는 홈 리스케줄링과 라인경합관리간의 역할분담 및 협조설계에 대한 기능 테스트를 목적으로 한다.

(나) 테스트 환경 구성

① 시험환경 구성

홈 리스케줄링 라인경합관리 연동테스트의 환경구성은 그림5-70과 같이 인접역 연동시험의 구성에서 라인정보 서브시스템과 라인경합관리 서브시스템을 추가하여 구성 하였다.

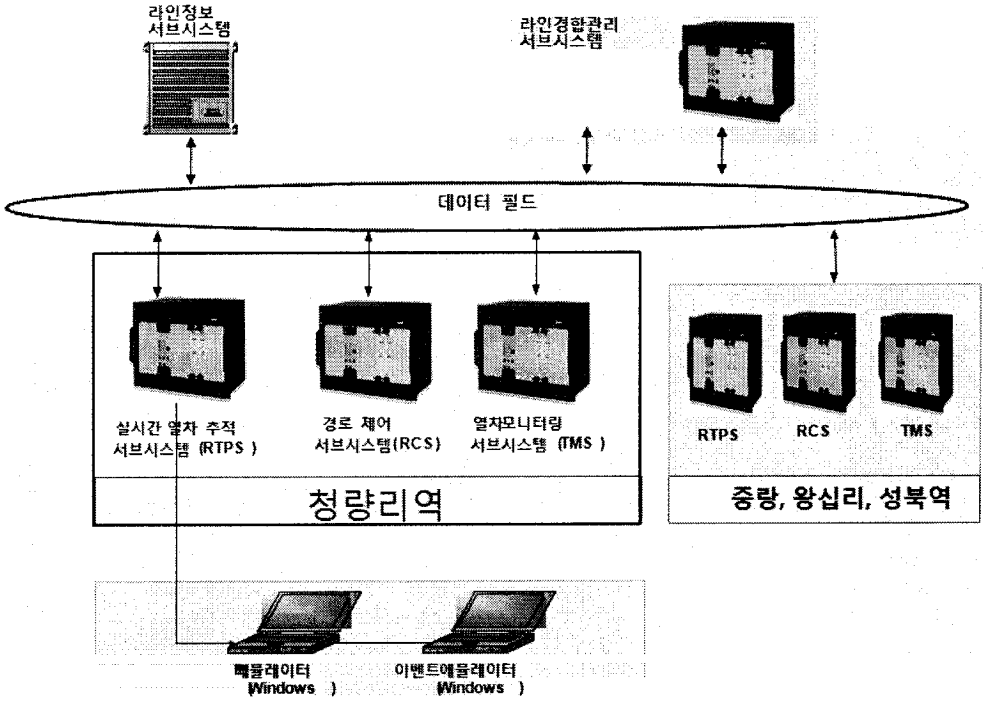


그림 5-70 라인경합관리 연동 테스트 시험환경 구성

② 테스트 스케줄 구성

시험에 사용되는 스케줄은 인접역 연동테스트에서 사용하는 청량리역 스케줄과 인접역 스케줄을 바탕으로 생성된 본선스케줄이다.

(다) 라인경합관리 연동 테스트 시나리오 설계

① HR 열차순서변경 제안 테스트

- HR 출발순서 고정지 열차순서 변경 제안

HR에서 출발순서를 고정시키는 경우 본선열차의 열차 출발 순서는 상하행, 라인별로 기존의 스케줄을 그대로 따르게 된다. 그러므로 라인경합관리로 열차순서 변경제안이 발생할 수 없다.

- HR 출발순서 변경시 열차순서 변경 제안

HR에서 출발순서를 변경가능 시키는 경우 홈 리스케줄링에서 역내 지연을 최소화하는 방향으로 본선열차의 열차 출발 순서가 상하행, 라인별로 기존의 스케줄에

서 변경이 발생할 수 있게 된다. 이때, 순서변경 제안 정보를 라인경합관리로 전송하여 라인관리자는 이 정보를 바탕으로 전체 라인의 열차 운영을 변경 하게 된다.

(라) 라인경합관리 연동 기능 테스트 결과

① HR 열차순서변경 제안 테스트

- HR 출발순서 고정시 열차순서 변경 제안

HR에서 출발순서를 고정시켜서 사용자 수동 경합해소를 하여 제어스케줄을 생성하였을 때, 새롭게 생성된 제어스케줄 상의 본선열차의 열차의 해당역 출발 순서는 상하행, 라인별로 기존의 결합스케줄상의 출발순서와 일치해야한다.

- 시험조건

테스트스케줄2, HR대상열차수 10, HR대상시간 30분, 삭제여유시각 60초, 경보여유시각 60초, 수정불가시각 2분, 라인별 출발순서고정

- 시험 절차 및 결과

표 5-24 HR 출발순서 고정 시험절차 및 결과

번호	입력 및 결과	비고
1	입력 : A0806열차의 대안진로 (출발:4H_B5002T)입력가능"적용"을 누른다.	
	결과 : 대안진로 입력 후 자동으로 경합해소(R1.dll)가 실행되어 스케줄이 변경되었을 경우 "경합해소 스케줄 생성 및 조회"화면이 뜨고 승인을 요구한다.	
2	입력 : 변경스케줄이 생성이 되고 "승인"버튼을 누른다.	
	결과 : 승인된 내용이 제어스케줄로 반영이 되었는지 확인한다.	라인별 출발순서 유지확인

② HR 출발순서 변경시 열차순서 변경 제안

HR에서 출발순서를 변경가능하게하고 사용자 수동 경합해소를 하여 제어스케줄을 생성하였을 때, 새롭게 생성된 제어스케줄 상의 본선열차의 열차의 해당역 출발순서는 변경이 가능하며 변경 발생 시에는 출발순서 변경 정보를 라인경합관리로

전송해야 한다.

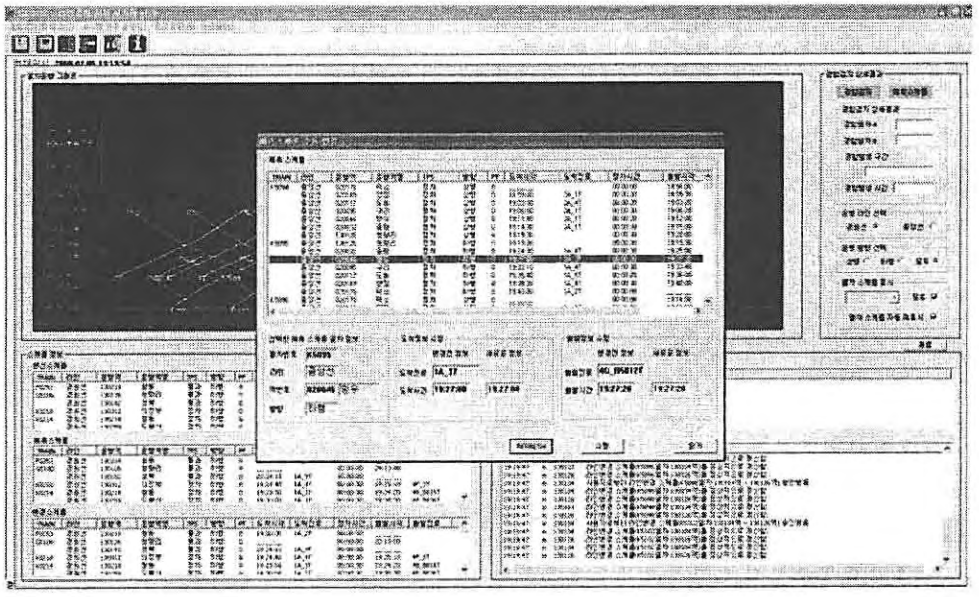


그림 5-71 출발순서변경 발생 경합해소

- 시험조건

테스트스케줄2, HR대상열차수 10, HR대상시간 30분, 삭제여유시각 60초, 경보여유시각 60초, 수정불가시각 2분, 라인별 출발순서변경

- 시험결과

표 5-25 HR 출발순서 변경 시험절차 및 결과

번호	입력 및 결과	비고
1	입력 : A0806열차의 대안진로 (출발:4H_B5002T)입력->입력가능->"적용"을 누른다.	
	결과 : 대안진로 입력 후 자동으로 경합해소(R1.dll)가 실행되어 스케줄이 변경되었을 경우 "경합해소 스케줄 생성 및 조회"화면이 뜨고 승인을 요구한다.	
2	입력 : 변경스케줄이 생성이 되고 "승인"버튼을 누른다.	출발순서 변경여부확인
	결과 : 승인된 내용이 제어스케줄로 반영이 되었는지 확인한다.	

여 백

6장 현장시험시스템 구축 및 시험

1절 현장시험 시스템 구축

개발된 대형·고밀도운행 철도역 자동운행제어시스템은 주관연구기관에 설치된 종합연계시험센터에서 기능시험 및 연계시험을 완료하였으나 실제 역 현장의 정보 및 열차운행정보를 얻을 수 없으므로 역 및 열차운행정보는 시뮬레이터를 연결하여 시험을 수행하였다.

개발된 시스템의 실제시스템과의 연계를 통하여 기능오류를 밝혀내고 보다 신뢰성 있는 시험을 위해 현장 시험을 실행하였다. 현장 시험 장소는 운행 중인 열차의 안전을 확보하기 위하여 한국철도공사와의 협의를 거쳐 오송역-청주역-오근장역을 선정하였다.

1. 현장 시범시스템 구축을 위한 사전 조사

가. 현장 조사

현장 조사는 오송, 청주, 오근장역에 대해 총 3회에 걸쳐 이루어 졌다. 현장 조사의 목적은 시험 장비의 설치 장소 섭외와 기존의 현장역의 장비와의 연결 등을 조사하기 위함이다.

그림은 6-1은 오송역의 열차 진로를 설정하는 역 담당자 있는 조작반의 환경이며, 그림 6-2는 오송역 열차운행 상태를 표현하는 선형표시 화면을 조사한 것이다.



그림 6-1 오송역 조작반

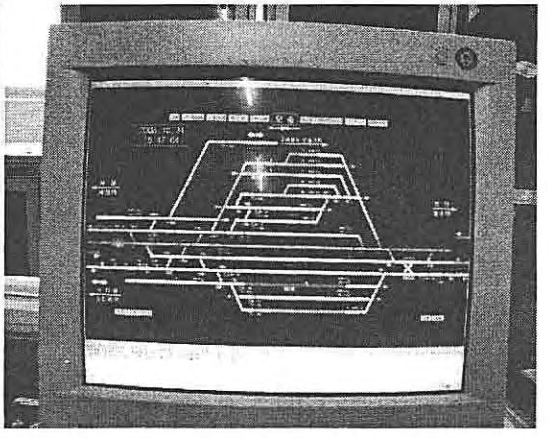


그림 6-2 오송역 선형표시

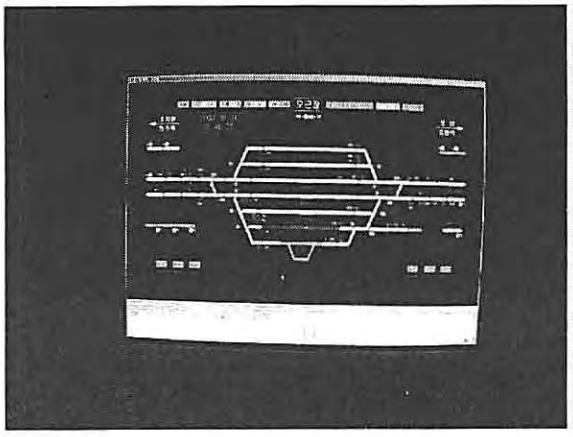


그림 6-3 오근장역 선형표시



그림 6-4 오근장역 LDTS



그림 6-5 청주역 조작반

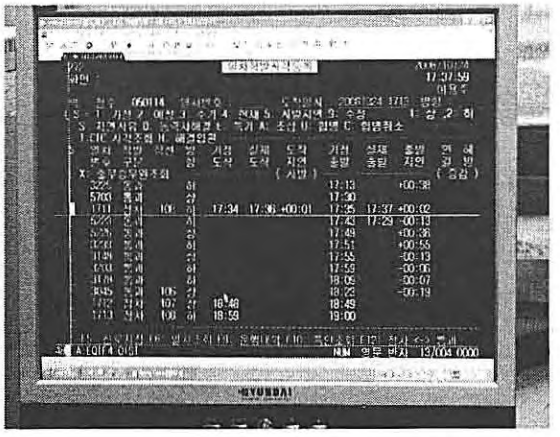


그림 6-6 청주역 KROIS

그림 6-3은 오근장역의 선형표시 화면이며 그림 6-4는 오근장역의 CTC 장비인 LDTS 장비이다. 개발된 자율분산 역제어시스템 시험을 위해서 LDTS 장비에서 통신라인 연결하여 시험하였다. 그림 6-5는 청주역 조작반 환경을 조사한 것이며 그림 6-6은 현재 열차 운행의 스케줄을 관리하고 있는 KROIS(Korea Rail Operation Information System)의 화면이다. 즉, 열차운용원은 선형표시 화면의 열차 점유상태와 KROIS의 열차스케줄을 일일이 대조하며 열차 운전승무원과 통신하여 열차번호를 대조하고 진로를 제어한다.

나. 통신라인 조사

대형·고밀도운행 철도역 자동운행제어시스템의 전체 기능 시험을 위해서는 역간 통신망 구축이 필수적이다. 역간 통신망 구축을 위해 철도공사에 기존에 구축되어

있는 철도 초고속통신망을 사용할 수 있도록 요청하였으며, 만일 철도공사가 충분한 회선을 갖추고 있지 못하는 경우에 대비하기 위하여 사설 전용망을 확보하도록 사전에 계획하였다. 철도공사의 협조에 의해 역간 광 통신 케이블 사용 가능하게 됨에 따라 역간 네트워크 구성을 위한 사전 현장 조사를 실시하였다.

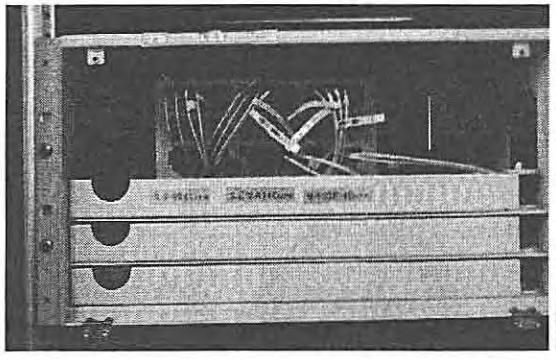


그림 6-7 통신 랙

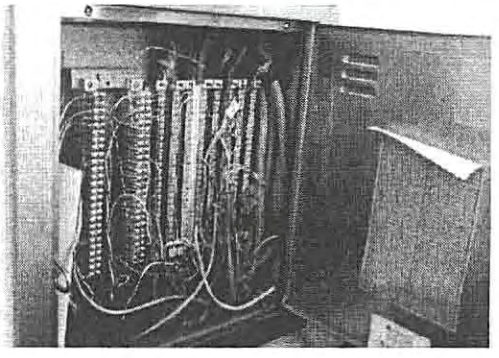


그림 6-8 통신 단자함

위의 그림 6-7과 6-8은 통신 랙과 통신 단자함으로 3개역에 동일한 통신 랙과 통신 단자함이 이미 설치되어 있다. 시험하게 될 3개 역의 자율분산 역제어시스템은 아래의 통신라인과 연결하여 시험을 하였다. 아래 그림 6-9는 3개역의 통신라인 연결 구성도이다.

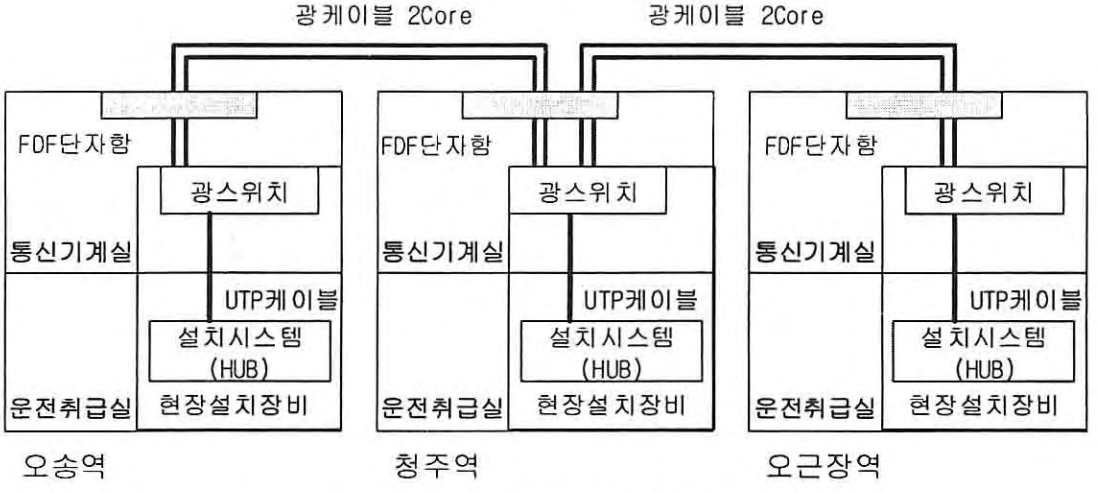


그림 6-9 역간 통신라인 구축

2. 현장 설치

가. 시스템 구성도

아래 그림 6-10에서 각 역의 연동장치와 연결된 LDTS는 역에 이미 설치되어 운영 중인 기존 시스템이며 LDTS와 연결된 자율분산 역제어시스템(ASCS)이 그 시험 대상 시스템이다. 각 시스템은 역간 네트워크로 연결되었으며 역당 광케이블 1회선으로 전송속도는 2M bps이다. LDTS와 ASCS의 전송속도는 9,600bps이며 케이블 규격은 0.9㎟(SQ)이다.

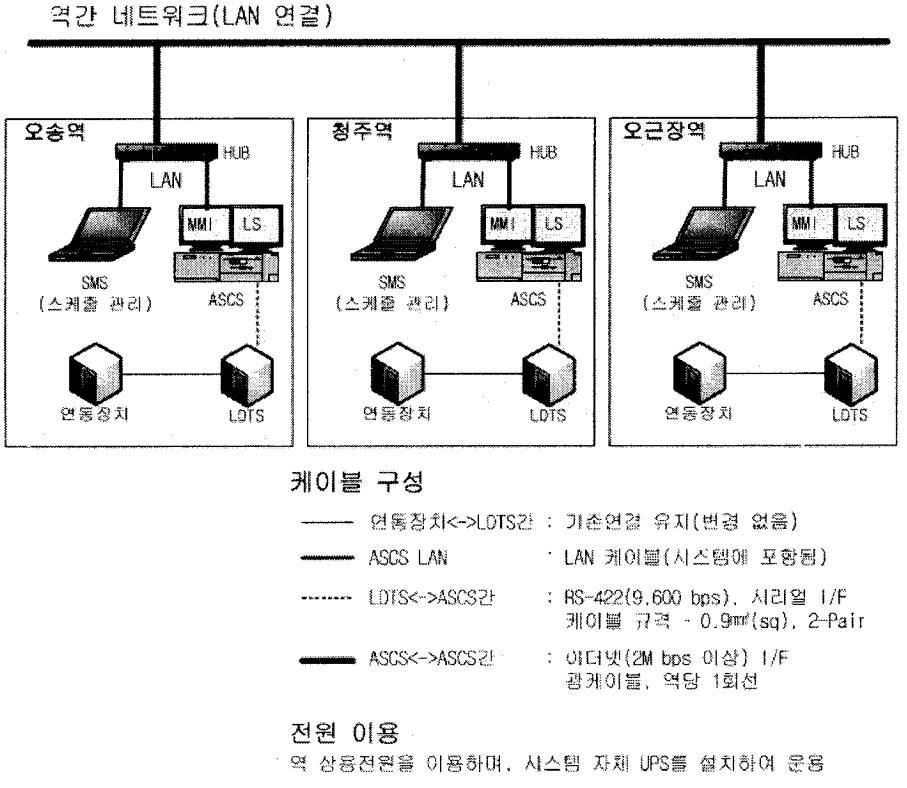


그림 6-10 현장 설치 시스템 구성도

나. 현장 장비 설치

각 역에 설치된 시스템은 그림 6-11과 같이 역 당 1set씩 설치되었으며 상세한 내용은 다음과 같다.

- 19" 표준랙 1EA
- UPS(Rack 타입): 1 SET

- 산업용 컴퓨터 : 1 SET
- 스위칭 허브 : 1 EA
- 19″ 모니터 : 2 EA
- 키보드 및 마우스 : 각 1EA

시험 시스템의 현장 장비 설치는 역간 통신라인 구축과 병행하여 시작되었다. 시험 시스템과 LDTS와의 인터페이스 이전에 LDTS ROM(Read Only Memory)교체 및 시험이 진행되었으며, LDTS의 동작이 이상이 없다는 것을 확인 한 후 시험장비와 LDTS간의 인터페이스가 구축되었다.

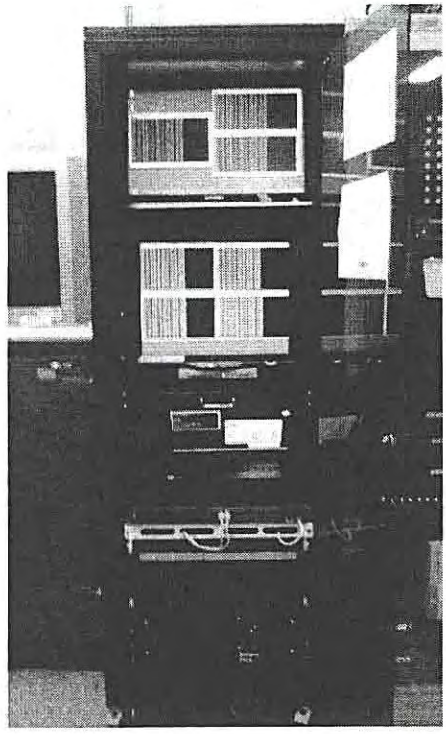


그림 6-11 설치된 시험시스템

그림 6-12는 오송역에 시험 시스템을 설치한 사진이며 그림 6-13 오송역 LDTS 단자 연결은 오송역 시험시스템과 LDTS 단자를 연결한 사진이다. 그림 6-14와 6-16은 오근장역과 청주역에 자율분산 역제어시스템이 설치된 사진이다. 그림 6-15와 6-17은 오근장역과 청주역에 설치된 시스템의 열차모니터링 서브시스템 화면과 수

송관리 서브시스템의 스케줄 화면이다.

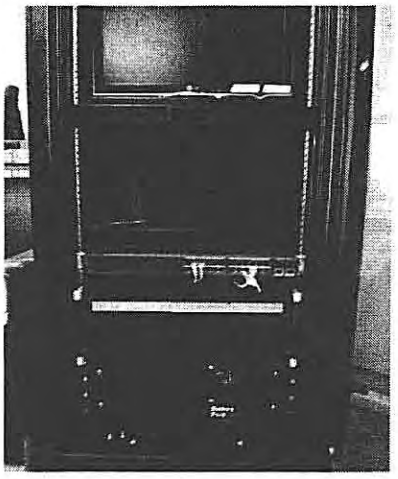


그림 6-12 오송역 설치 후

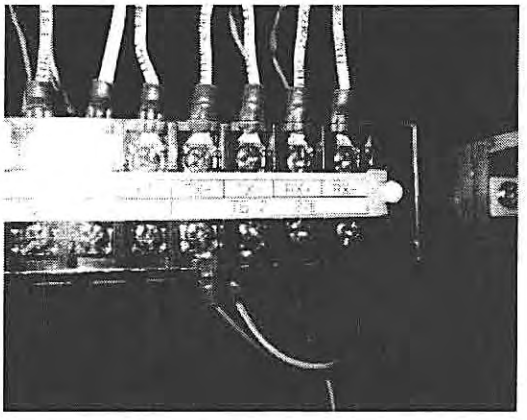


그림 6-13 오송역 LDTS 단자 연결

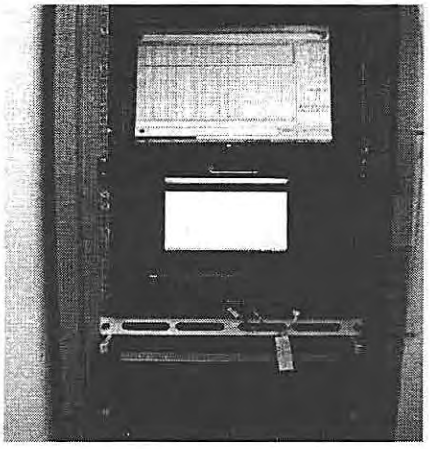


그림 6-14 오근장역 설치 후

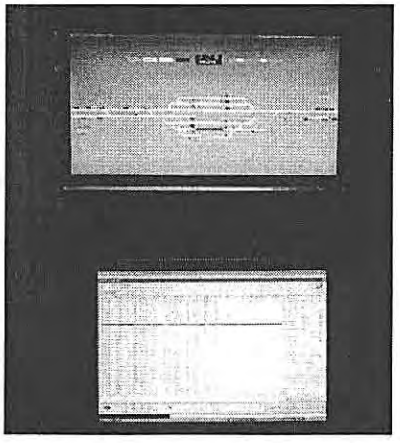


그림 6-15 오근장역 운영



그림 6-16 청주역 설치 후

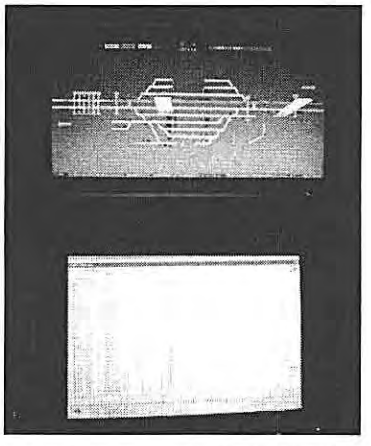


그림 6-17 청주역 운영

2절 현장 시험

1. 자율분산 시스템 속성 시험

가. 시험 목적

자율분산 역제어시스템은 자율분산시스템 기술을 적용하여 개발되어 졌다. 자율분산 시스템은 하드웨어나 소프트웨어를 구축하는 것뿐만 아니라 건설에서 운용까지 포함한 시스템 구축의 사고방식의 개념을 가진다. 시스템 구축에 투입되는 비용보다 시스템 운용에 관계되는 비용이 커짐에 따라 시스템의 운용에 관계되는 조직이나 사람의 의존도가 커진다. 특히 시스템의 규모가 크고 경제적으로나 사회적 영향이 큰 시스템이 그러하며 이러한 시스템 운용에 대한 요구사항이 생기기 시작했다.

시스템의 상황에 맞추어 시스템을 중지 없이 변경하거나 수정하고 단계적으로 건설할 수 있는 온라인 확장의 요구사항과 시스템 일부를 확장하거나 장해부분을 수리복구 하였을 때 점검이 필요할 때 사용할 수 있는 온라인 유지보수 기능이 그 대표적이다. 이러한 요구사항을 반영한 자율분산 역제어시스템의 주요 속성은 자율제어성과 자율 협조성이다. 따라서 자율분산 역제어시스템의 자율 제어 시험과 자율 협조 시험을 수행하였다.

나. 시험 환경

세 개의 자율분산 역제어시스템이 청주역과 오근장역, 오송역에 설치되었다. 자율제어와 자율협조 시험을 위해서 세 개의 역 모두 활용되었다.

다. 시험 절차 및 결과

(1) 자율제어성 시험

(가) 시험 개요

자율제어성의 의미는 물리적으로 구성된 네트워크에 연결된 여러 대의 서브시스템 중에서 만일 하나의 서브시스템이 기능 불가동이 되어도 잔여 서브시스템은 스스로의 자율분산 역제어시스템을 제어할 수 있어야 한다는 것이다. 자율제어 시험

의 목적은 역시스템이 기능 불능상태일 때 나머지 역의 시스템이 자율적으로 기능을 수행하는지 테스트하였다.

(나) 시험 절차

- 인접역인 오근장역과 오송역이 연결된 상태에서 인접역 열차추적정보를 활용하여 자동진로제어를 수행한다.
- 인접역인 오근장역의 시스템이 기능 불능 상태로 변경한다.
- 단독 역 상태에서 정상적으로 운영되는 열차운행제어 상태를 시험한다.

(다) 시험 결과

시험절차에 따라 오송역과 오근장역의 열차추적 정보가 청주역 시스템에 수신되는지를 외부테스터에서 확인하였다. 아래 그림에서 오송역 번호인 50044와 오근장역 번호인 502154가 청주역에 설치된 외부테스터에 수신되는 것을 확인하였다. 즉, 인접역이 모두 연결 상태인 것을 확인하였다.

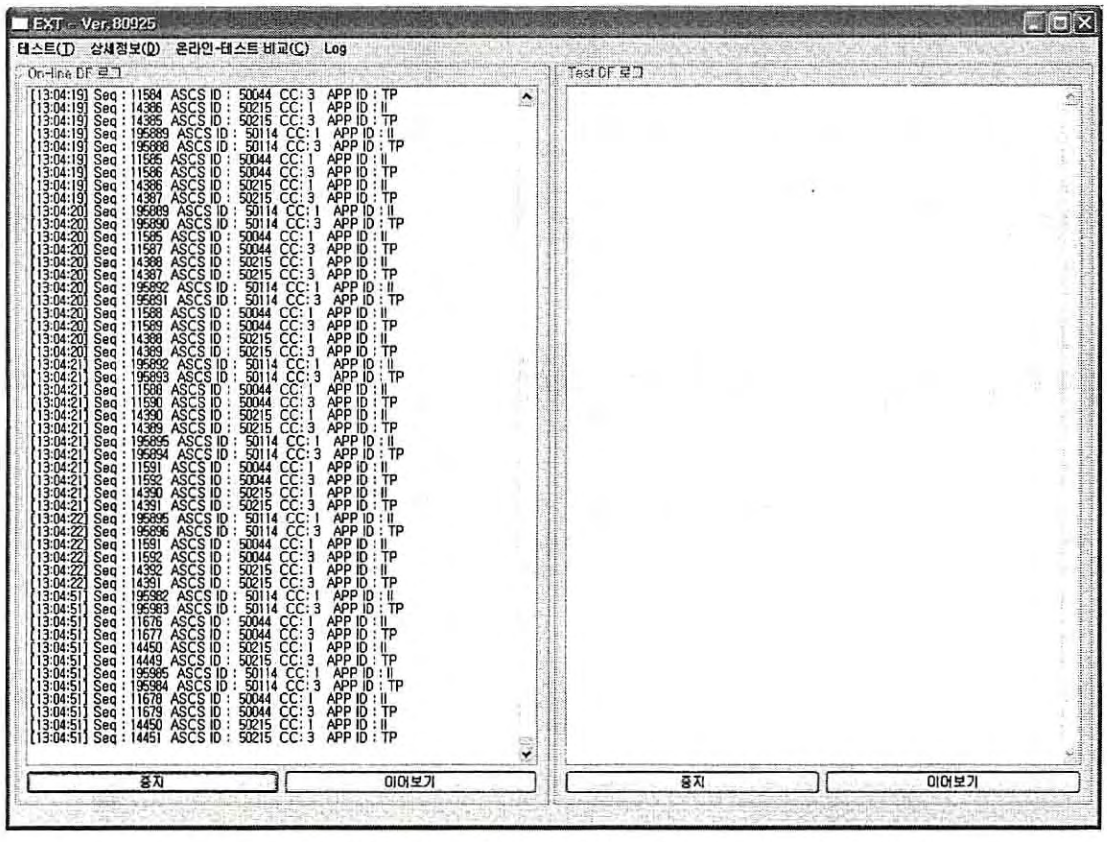


그림 6-18 청주역에서 오근장역, 오송역 정보수신 화면

시험을 위해 인접역을 의도적으로 다운 시킨다. 인접역이 다운된 사실은 아래 그림 6-19와 같이 외부테스터에서 인접역 데이터가 들어오지 않는 상태 메시지인 'NullPointerException' 정보로 확인하였다.

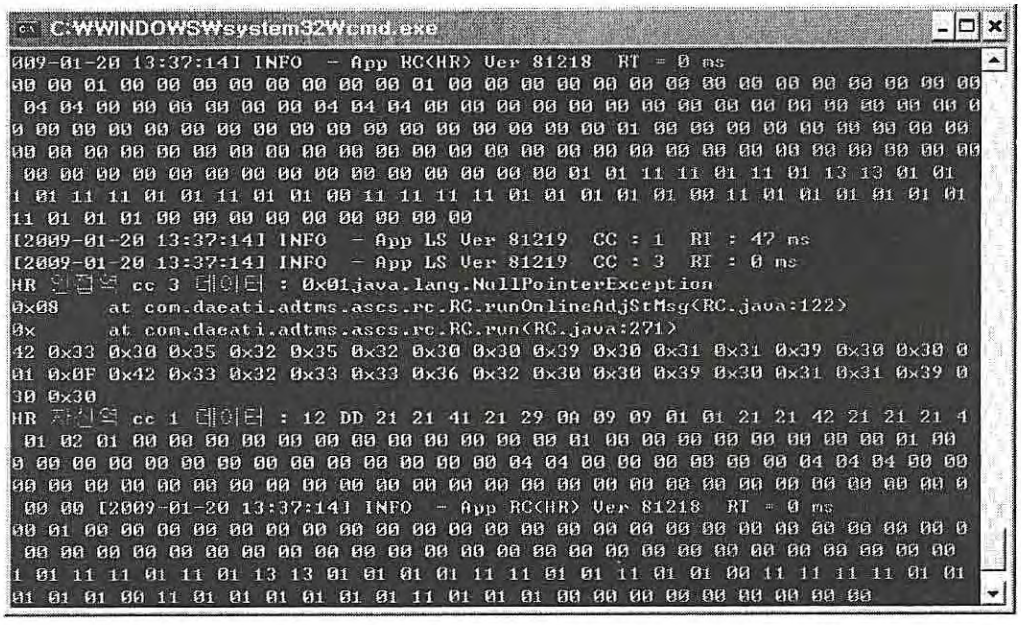


그림 6-19 인접역 데이터 수신 불능 상태 확인

이와 같이 인접역 정보가 없는 상태에서도 청주역의 진로가 1A에서 8T로 진로 제어 명령이 송출된 것을 아래 그림 6-20에서와 같이 알 수 있다.

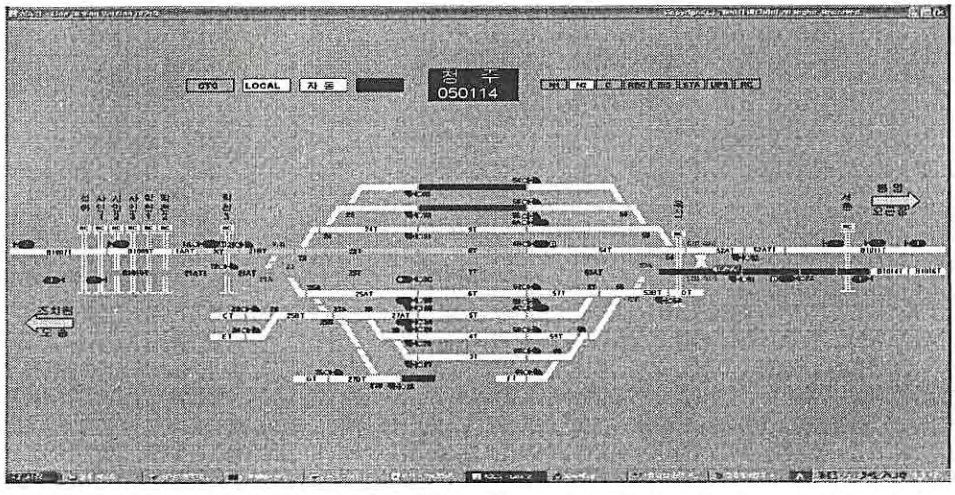


그림 6-20 진로제어 명령의 표시

또한 화면에 출력된 진로설정 정보를 명확히 확인하기 위해서 외부테스터에서의 진로제어 CC가 2번인 역제어 명령이 송출된 것을 확인하였다.

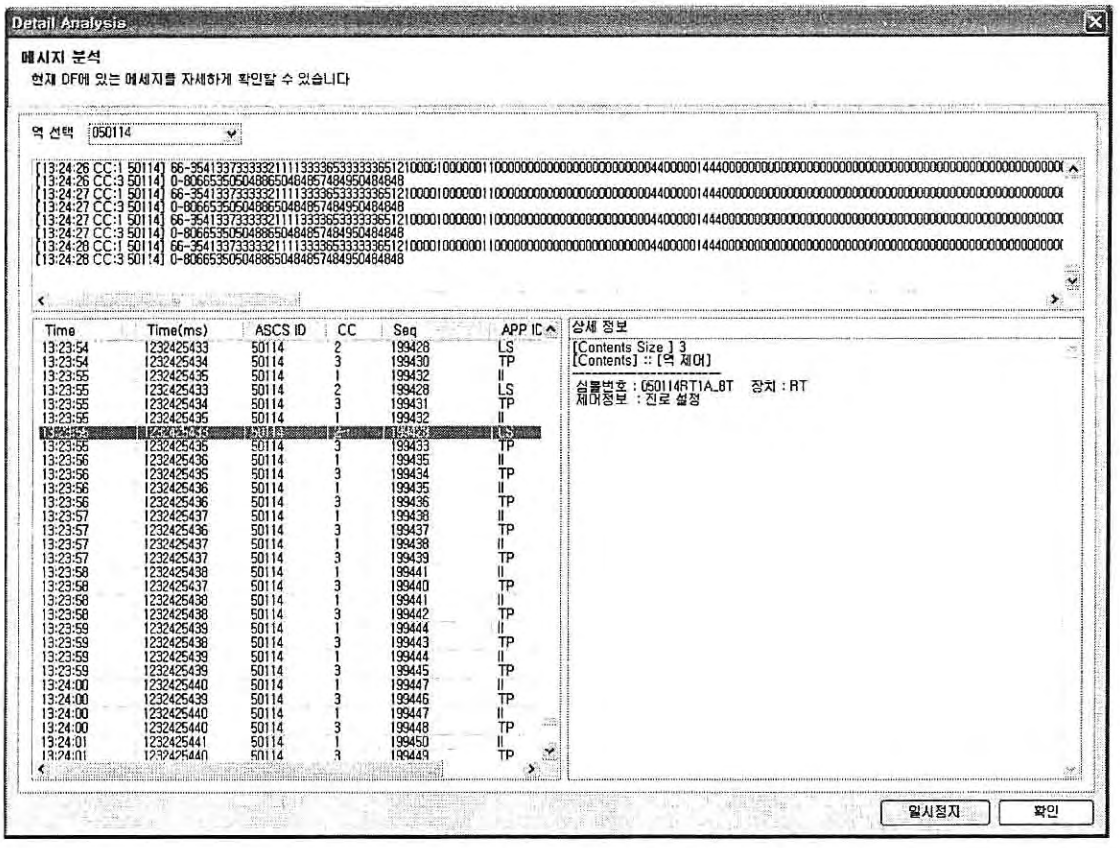


그림 6-21 외부테스터에서의 진로명령 출력 확인

자율제어성 시험은 오송역이 고장상태에서도 청주역이 고장에 영향을 받지 않고 정상적으로 가동하는 상황을 시험하였다. 즉, 하나의 시스템이 고장상태일 때 전체 시스템에 영향이 없다는 사실이 시험으로 확인하였다.

(2) 자율협조성 시험

자율협조성의 의미는 물리적으로 구성된 네트워크에 연결된 여러 대의 서브시스템 중에서 만일 하나의 서브시스템이 기능 불가동이 되어도 잔여 시스템간의 상호간 협조가 이루어질 수 있다는 것이다. 자율 협조 시험의 목적은 역시스템 하나가 기능불능 상태에서 나머지 두 역의 협조가 이루어지는지 시험하였다.

자율협조 시험은 아래 그림 6-22와 같이 오근장역을 정상으로 운행하다가 기능불능상태로 변환하여 청주역과 오송역 간의 협조 관계 분석 시험하였다.

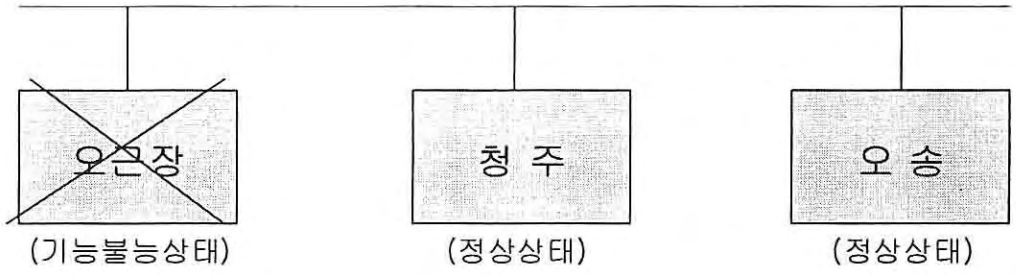


그림 6-22 자율협조 시험 상황 개요도

청주역에 설치된 외부테스트를 통해 오근장역(50215) 정보가 정상으로 수신되는 상황을 아래 그림 6-23을 통해서 확인하였다.

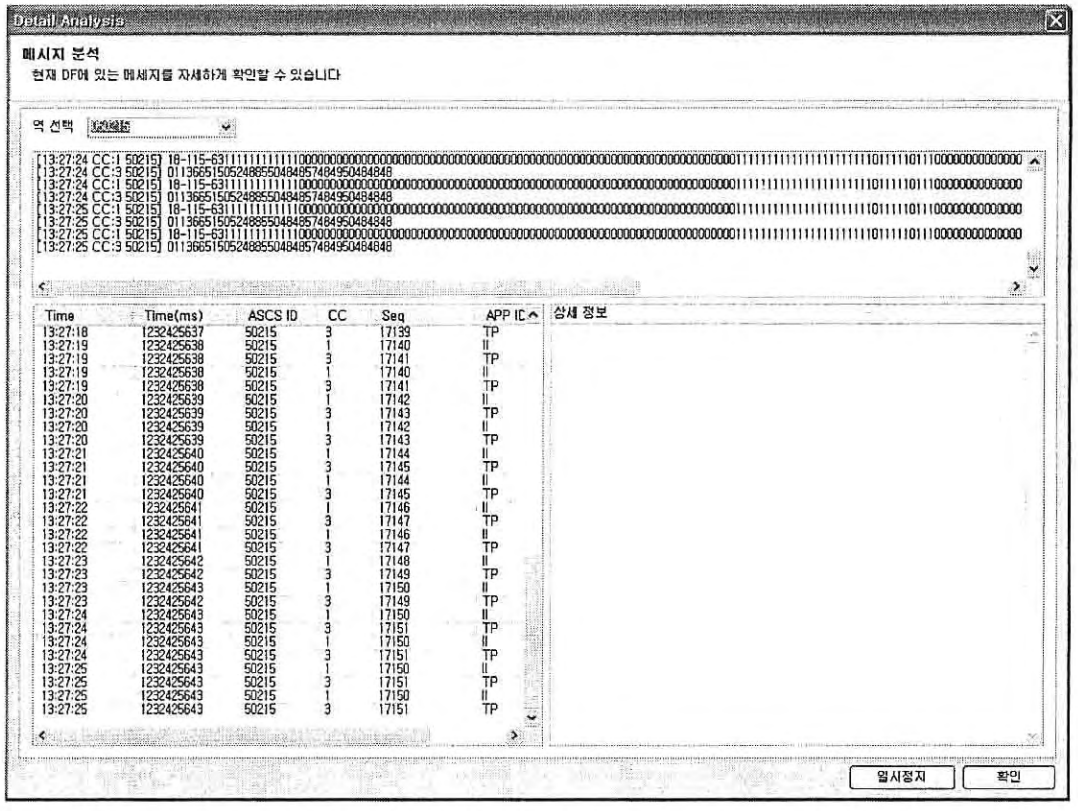


그림 6-23 청주역 외부테스터에서 오근장역의 정보 수신

청주역에서 위의 그림 6-23과 같이 모든 인접역 정보가 수신되는 것을 확인한 이후에 임의로 오근장(50215) 역을 기능불능상태로 변경한다. 이를 확인하기 위해 청주역 외부테스터에서 오근장역번호가 수신되지 않는 것을 그림 6-24와 같이 확인하였다. 이러한 상태에서 청주역은 오송(50044)역 정보를 이용하여 그림 6-25와

같이 정상적인 진로설정이 이루어지는 것을 확인하였다.

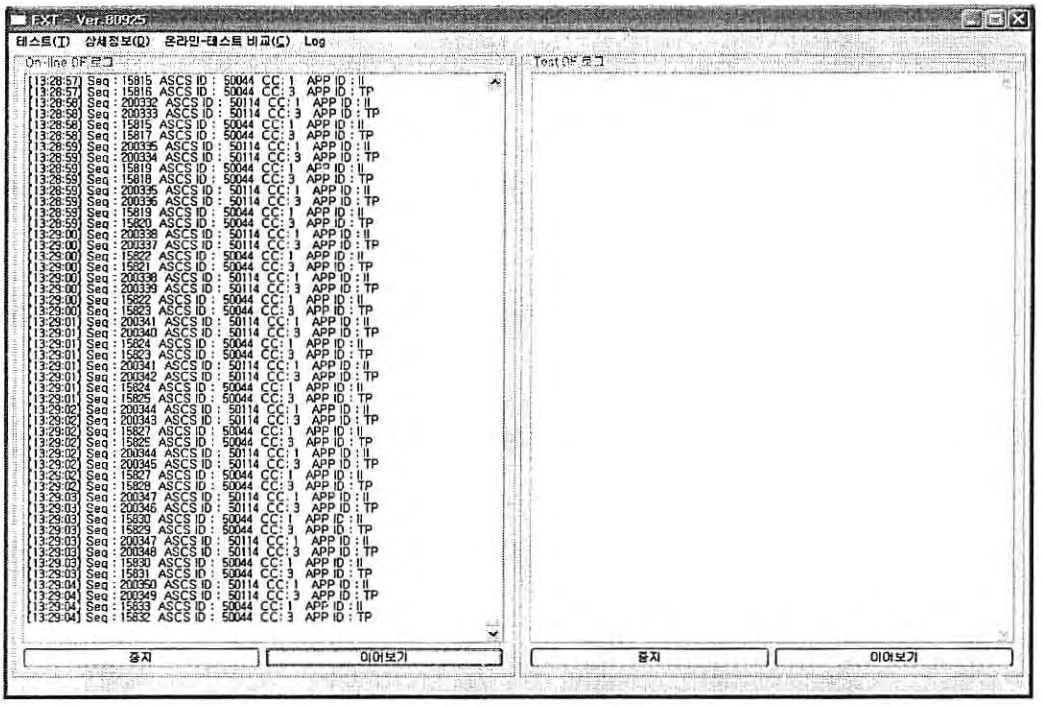


그림 6-24 청주역에서 오근장역 정보가 수신되지 않는 상태

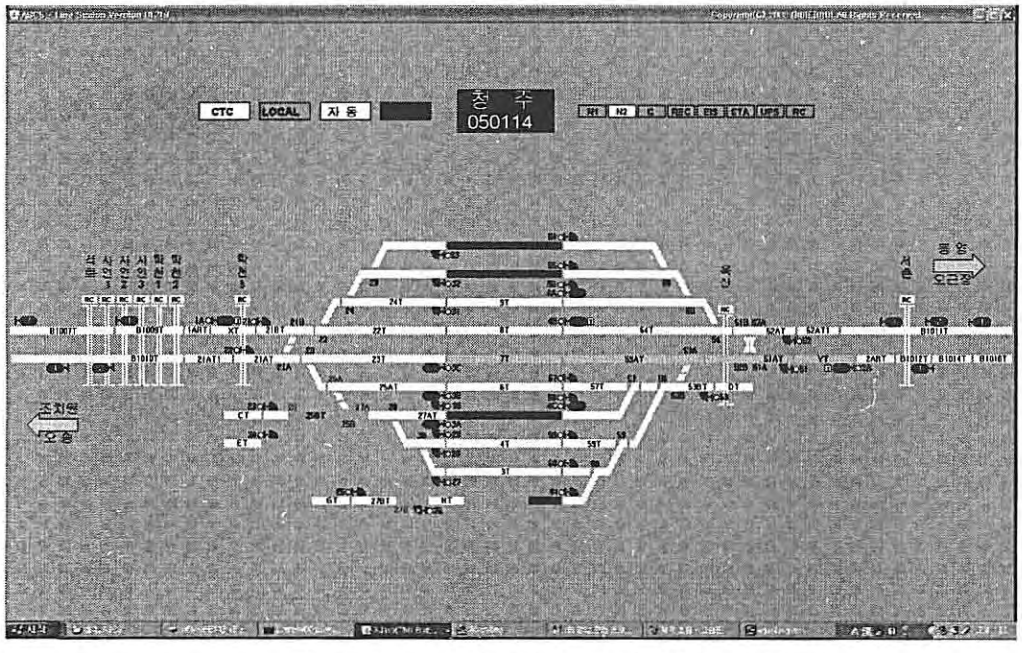


그림 6-25 청주역 열차 진로 설정 확인

2. 온라인 시스템 확장 시험

가. 시험 목적

열차 운행제어시스템과 같이 넓게 분산된 시스템은 오랜 시간동안에 구축이 필요하다. 장기간의 구축동안에 모든 서브시스템들은 온라인 운영의 정지 없이 확장되어져야 한다. 자율분산 역제어시스템도 하나의 역에 설치되어 운영되면서 그 제어영역을 점진적으로 확대할 수 있도록 설계되었다. 이러한 부분은 시스템의 운영모드를 온라인 운영 상태와 테스트 운영 상태로 구분하여 온라인 데이터와 테스트 데이터를 활용하여 새롭게 구축되는 시스템을 테스트 할 수 있다. 온라인 시스템 확장뿐만 아니라 온라인 유지보수의 필요성은 시스템의 고장 이후의 재 가동시의 테스트나 주기적인 유지보수 활동 등에 필요하다.

따라서 온라인 시스템 확장 시험의 목적은 하드웨어적인 역시스템의 확장이 경우를 설정하여 역 확장 절차에 대한 테스트 항목을 도출하고 도출된 테스트 절차에 맞게 시스템을 테스트 하는 것이다.

나. 시험 환경

온라인 시스템 확장의 단계는 단독 역을 테스트 하고 인접역으로 확장되는 경우를 테스트 하며 최종적으로 센터시스템과의 테스트를 통하여 새롭게 구축되는 시스템을 온라인으로 실시간 구동시키게 된다.

테스트 현장에서는 단독 역 테스트와 인접역으로 확장하는 테스트까지 실시하였고 센터 연동 테스트는 종합연계시험센터에서 그 내용을 테스트 하였다. 온라인 확장 시험은 청주역과 오근장역을 활용하여 테스트 하였다.

다. 시험 절차 및 결과

(1) 단독역 테스트

단독역 테스트는 청주역에서 테스트를 수행하였다. 테스트 내용은 LDTS 연동테스트, RTPS 열차 추적 테스트, 자동 및 수동 진로제어 테스트, 홈 리스케줄링을 통한 스케줄 변경 테스트, TMS를 활용한 GUI(Graphic User Interface) 테스트를 수행하였다.

시험 환경은 아래 그림6-26과 같이 개발한 자율분산 역제어시스템(ASCS)을 현재 역시스템에 구축되어 있는 LDTS와 연결하였으며 제어데이터를 현장의 안전을 위해 물리적으로 절체 되어 있다.

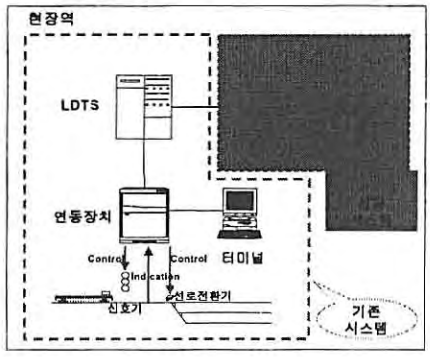


그림 6-26 단독역 테스트 구성도

(가) ASCS와 LDTS 연동 시험

자율분산 역제어시스템이 단독으로 역에 설치될 때 첫 번째로 테스트 하는 것이 LDTS나 연동장치와 연결하여 역 상태정보 수신을 확인하는 것이다. 역 상태 정보 수신 확인을 위해 LDTS와 연결된 청주역 ASCS에서 외부테스터를 연결하여 역 표시정보 수신을 확인하였다. 아래 그림6-27에서 청주역에서 내용코드(CC)가 1번인 역 표시정보가 표출되는 것을 확인하였다. 또한 열차모니터링 서브시스템의 화면에서도 역 표시정보를 반영한 열차의 점유 정보 및 진로가 설정된 상황을 확인하였다.

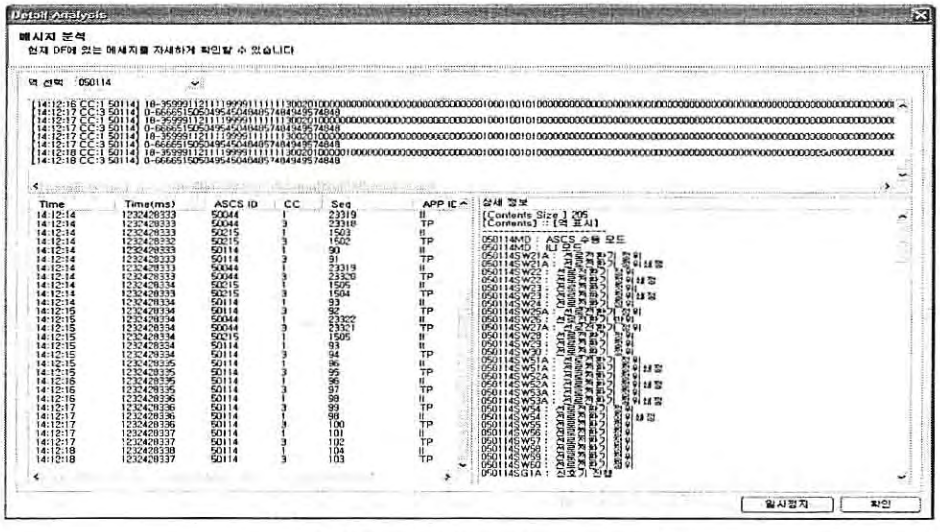


그림 6-27 LDTS 연동시험 결과

위의 그림 6-27은 외부테스터에서 LDTS에서 수신된 현장 정보를 확인하였고 그 내용이 TMS에서 표시되는 것을 그림 6-28로 확인하였다.

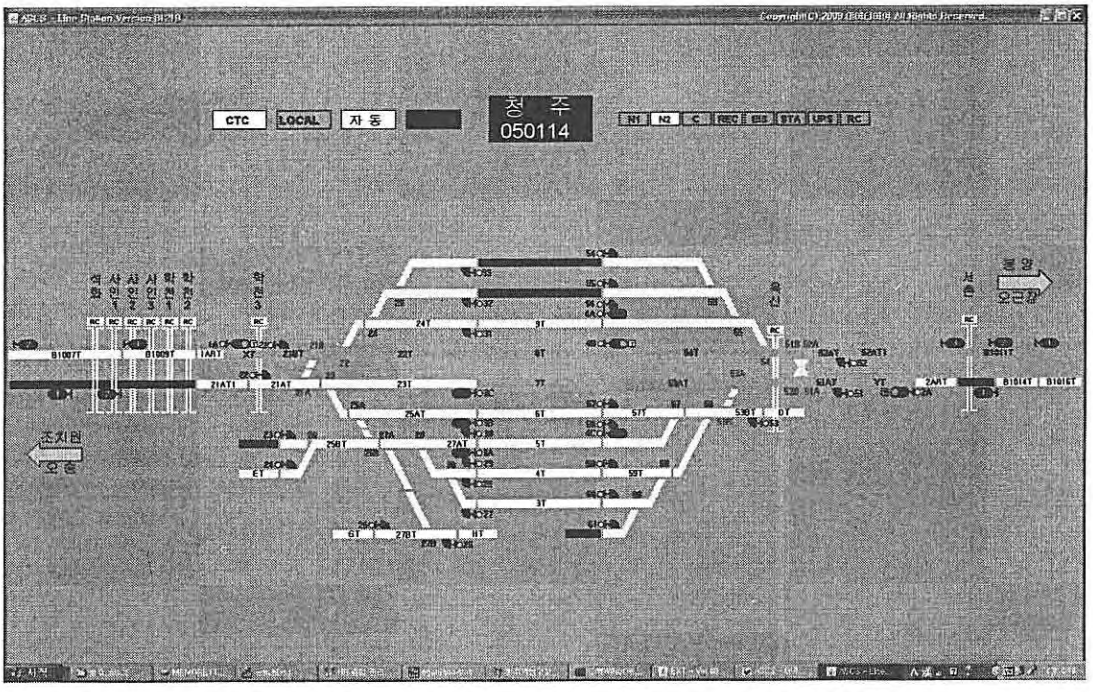


그림 6-28 현장 정보 수신에 의한 TMS 화면 표시 결과

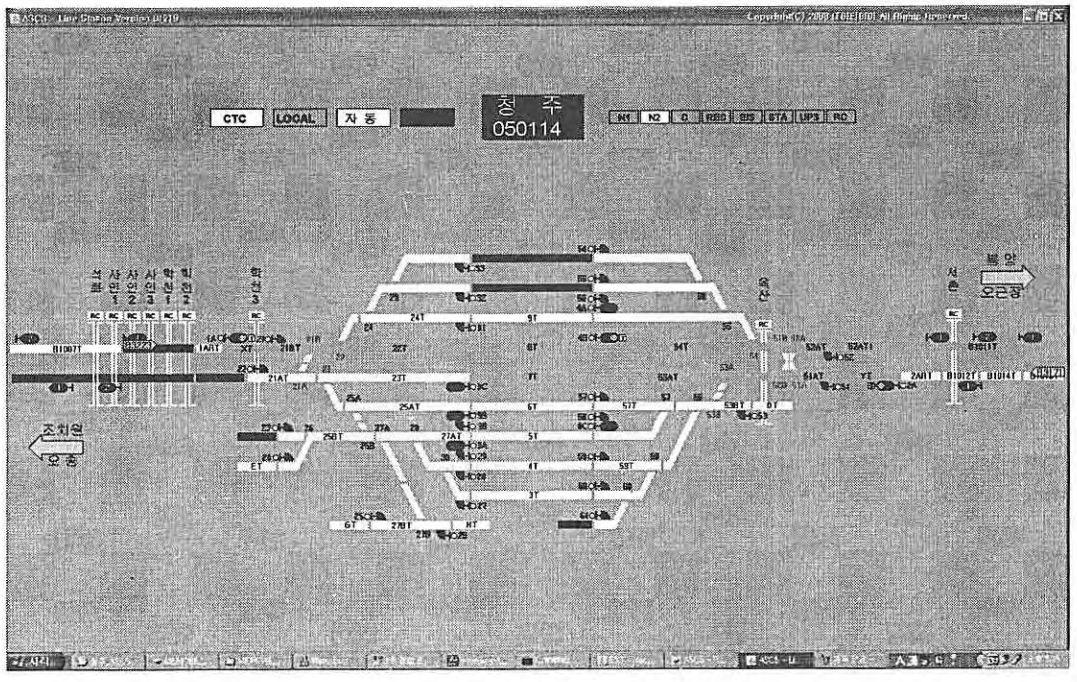


그림 6-29 열차추적 시험 결과

(나) 열차추적

열차추적 시험은 열차모니터링 시스템의 선형 화면에 열차 점유정보에 열차번호가 출력되는지를 확인하는 것으로 시험을 하였다. 위의 그림 6-29에서 주 선로 좌측에는 B3223 열차와 우측에 B3171 열차가 점유되어 있는 것을 확인할 수 있다.

(다) 진로제어

진로제어 명령이 올바르게 수행 되는지 확인하기 위해 청주역의 53BT에서 6T로의 진로가 설정된 것을 아래 그림 6-30에서 확인하였다.

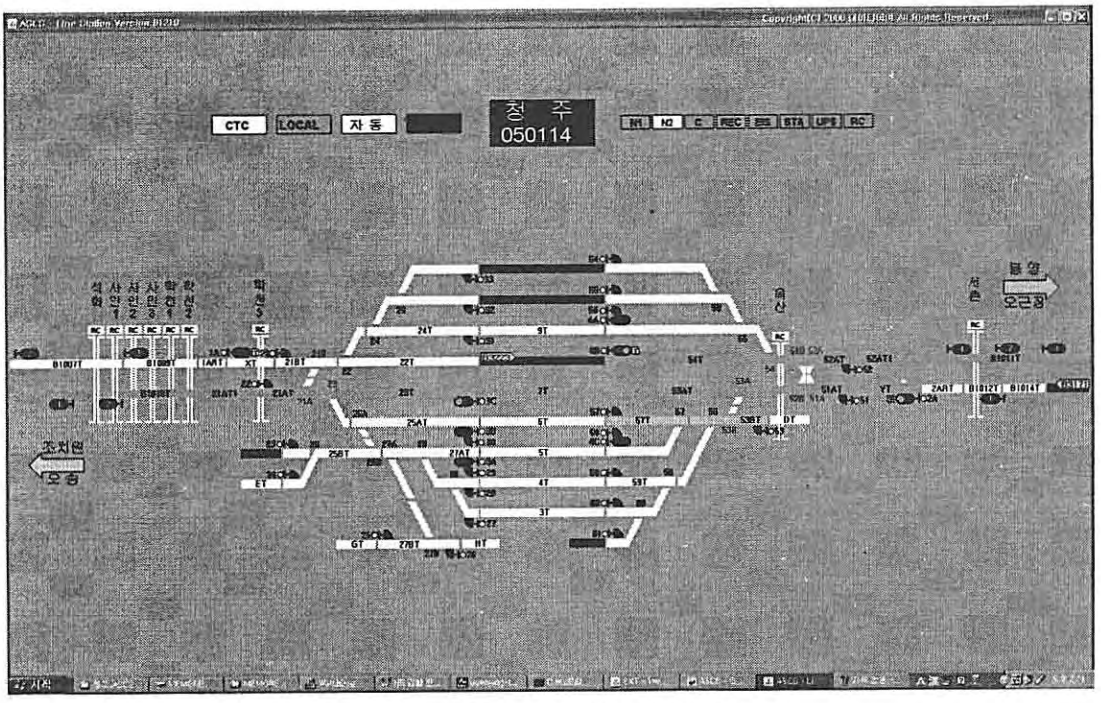


그림 6-30 진로제어 시험 결과

또한 이 진로제어 명령이 출력되었는지를 외부테스터를 통해 확인할 수 있었다. 아래 그림 6-31에서와 같이 청주역(50114)에서 내용코드(CC) 번호가 2번인 역제어 명령이 출력된 사항을 확인할 수 있었다.

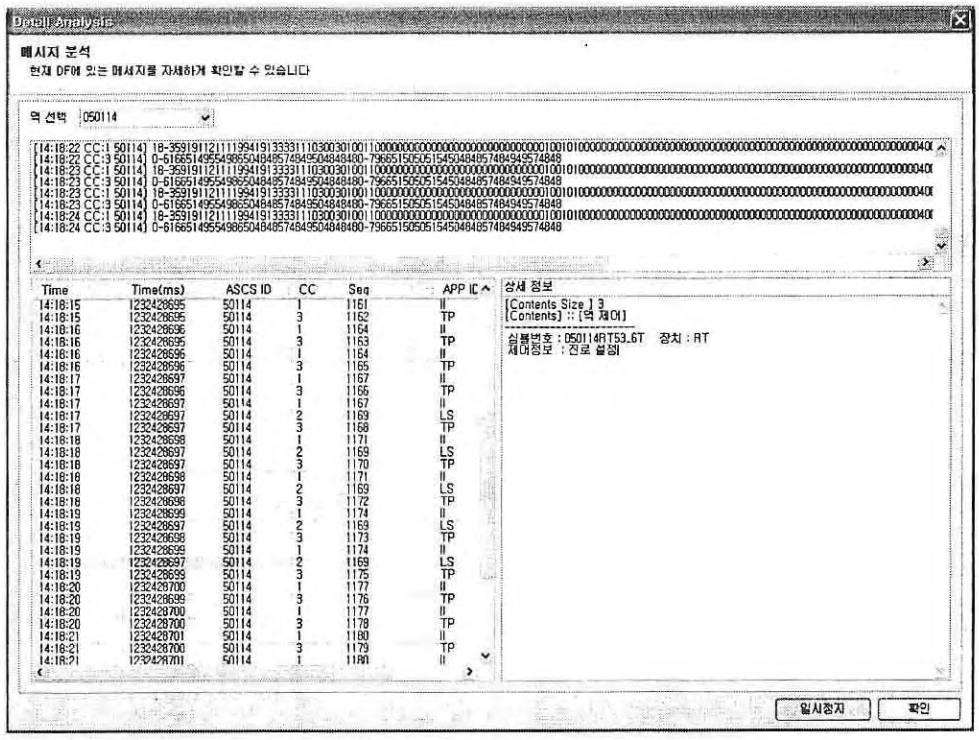


그림 6-31 외부테스터를 활용한 제어정보출력 결과 확인

(2) 인접역 확장 테스트

인접역 확장 테스트 환경은 온라인으로 구동되는 오송역과 인접한 역인 청주역이 확장되는 단계를 테스트한다. 테스트 절차는 아래 그림 6-32와 같은 단계를 거친다. 인접역인 오송역을 온라인 모드로 운영하고 역시스템을 확장하려고 하는 역인 청주역을 테스트 모드로 설정한다. 청주역의 자율분산 역제어시스템에서 인접역에서 송신한 데이터가 수신되는지 확인한다. 인접역 정보가 수신되면 인접역에서 송신한 열차출발 정보를 활용하여 역에서 진로제어를 수행하기 때문에 수신한 열차출발 정보에 따라 역 진로를 설정하는 것을 테스트 하였다.

그림 6-33은 청주역과 오송역간의 연결 상황을 나타낸 것이다. 이 그림에서 역 A는 온라인으로 운영 중인 오송역의 상황을 나타낸 것이며 역 B는 새롭게 확장할 청주역의 상황을 개념도로 표현하였다.

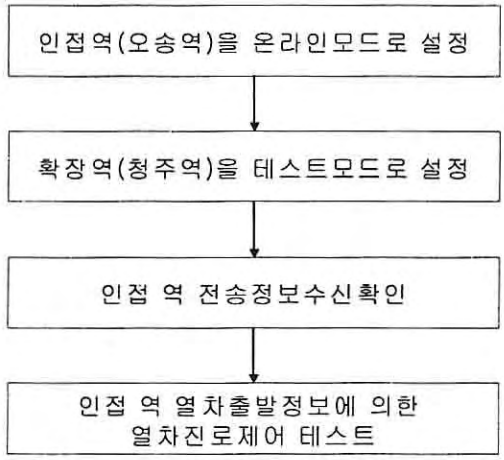


그림 6-32 온라인 확장 테스트 절차

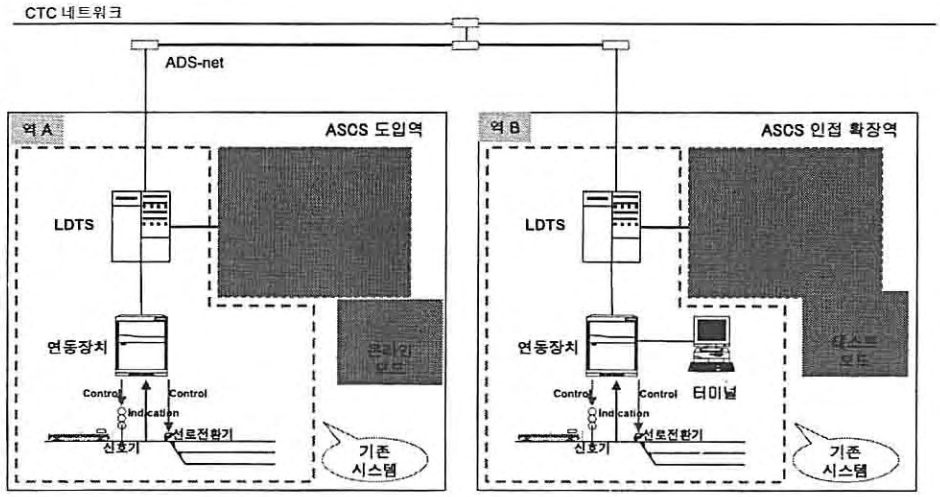


그림 6-33 인접역 확장 시험 개념도

(가) 인접역(오송역) 온라인 모드 설정

온라인 시스템 확장을 위해 오송역을 온라인으로 구동한다. 청주역의 외부테스터에서 오송역의 온라인 정보가 수신되는지를 아래 그림 6-34에서 확인하였다. 이 그림에 3개의 창이 있는데 상단의 창에 표시되는 로그정보에 'Test-Mode'라는 정보가 있으면 수신되는 내용이 테스트 데이터이다. 그림 6-34에서는 이러한 정보가 없기 때문에 온라인 데이터이라는 것을 확인할 수 있다. 또한 청주역에서 수신되는 오송역의 온라인 데이터가 역 번호인 50044를 통해 역표시 정보(CC=1)와 열차번호

표시(CC=3)가 수신되는 것을 확인하였다.

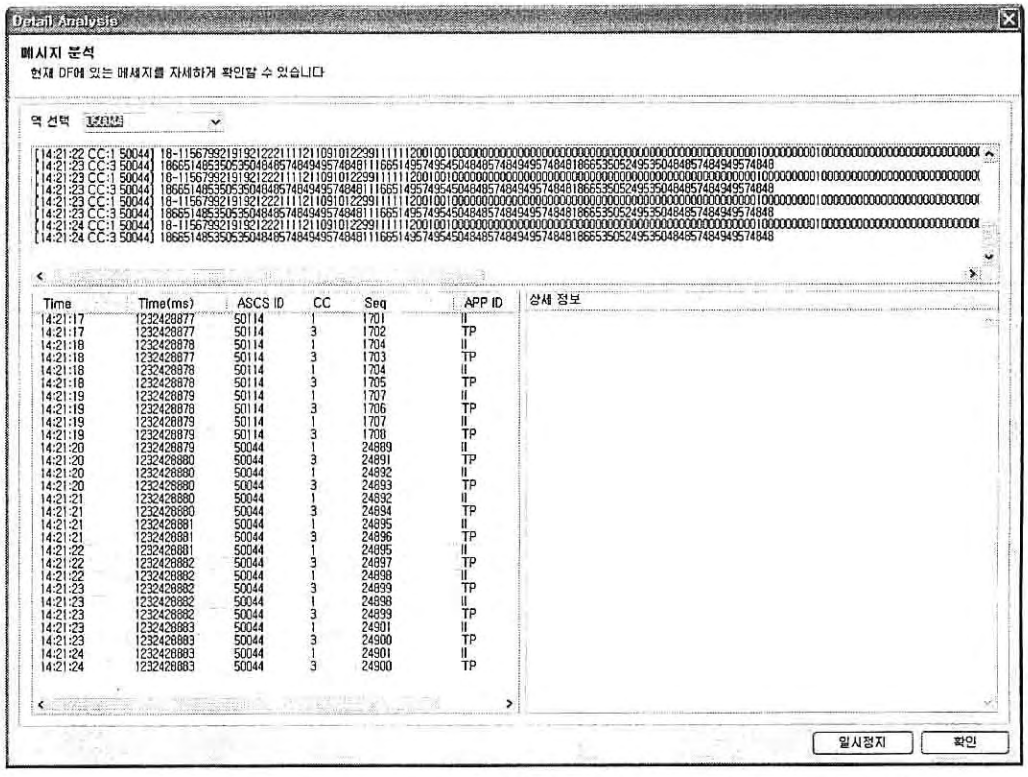


그림 6-34 온라인 확장을 위한 인접역 온라인 구동 확인

(나) 확장역 테스트 모드 설정과 인접역 정보수신 확인

새로 확장하려는 청주역을 테스트 모드로 시험하기 위해 그림 6-35와 같이 청주역 자율분산 역제어시스템의 자율분산관리자 모드를 테스트모드로 설정하였다.

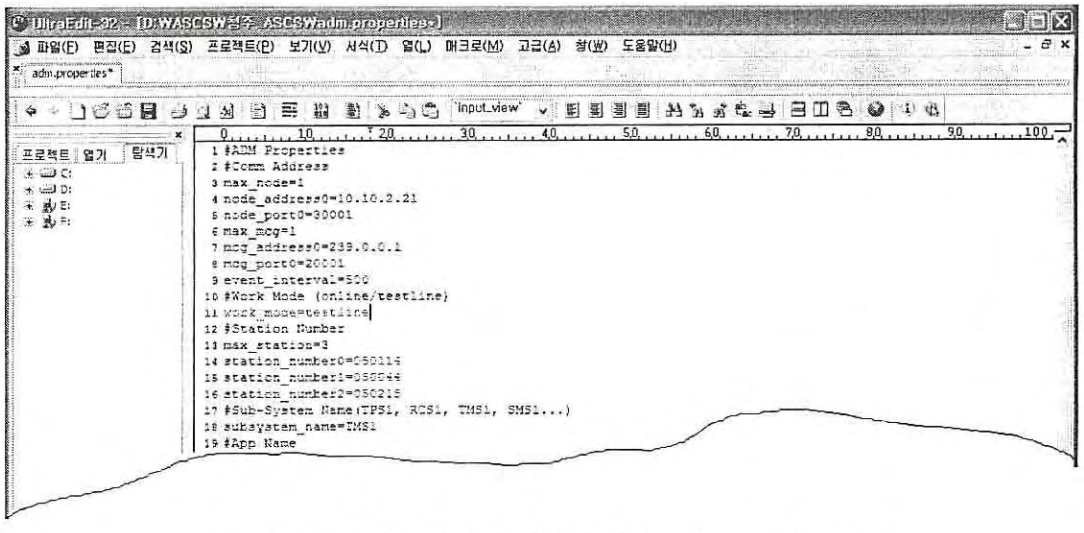


그림 6-35 자율분산관리자에서의 테스트모드 설정

모드 설정 후 시스템을 구동한다. 외부테스터에서 테스트모드인 청주역 데이터와 온라인모드인 오송역의 데이터가 수신되는지 확인한다. 그림 6-36과 같이 각 역에서 송출하는 온라인과 테스트 데이터가 수신되는 것을 볼 수 있다. 그림 6-37에서는 상세한 오송역의 열차번호와 점유대수를 확인할 수 있다.

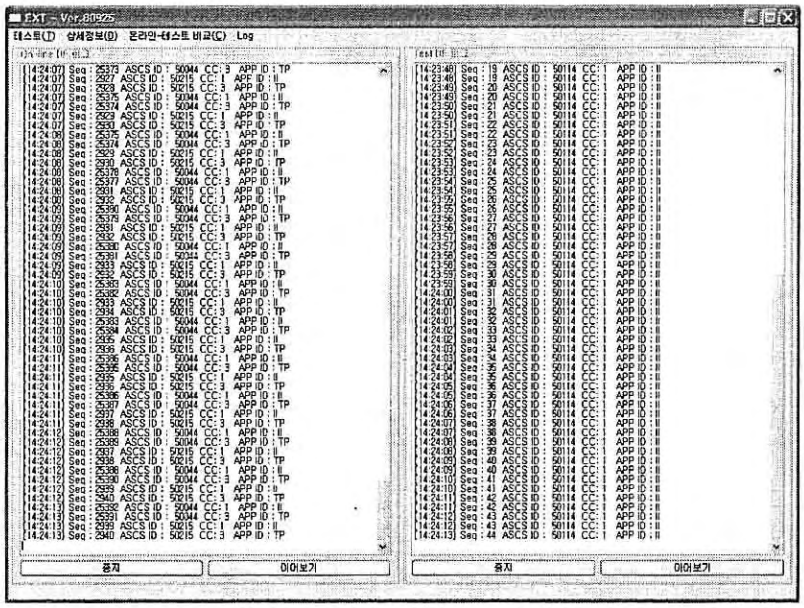


그림 6-36 온라인데이터와 테스트데이터의 수신확인

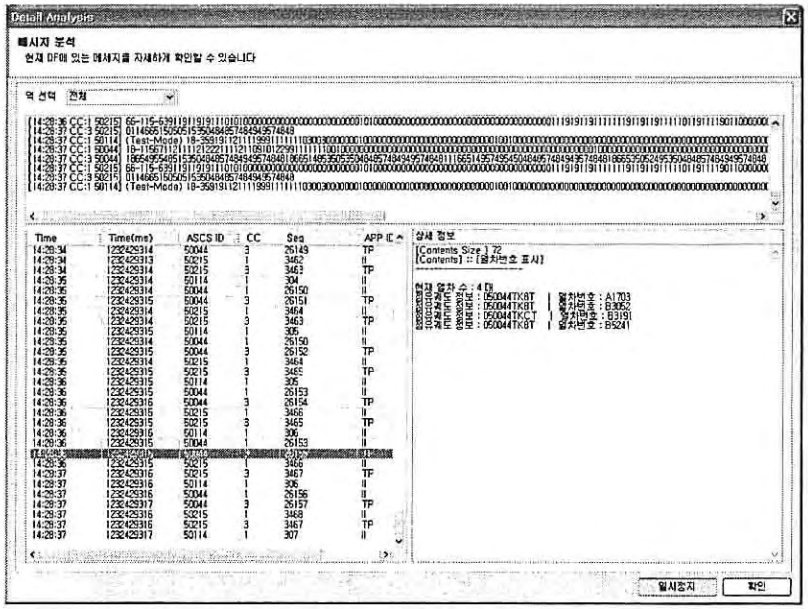


그림 6-37 오송역 열차출발정보 확인

이와 같이 확장하는 청주역에서는 테스트 모드 상태에서 온라인으로 송출되는 오송역의 온라인 데이터인 열차번호정보(CC=3)를 활용하여 진로를 설정하게 된다.

(다) 열차진로제어 테스트

역시스템의 확장 테스트를 통해 테스트 모드 시스템에서 온라인 시스템의 데이터를 수신하는 것까지 시험하였다. 온라인 확장의 마지막 단계 시험은 인접역인 오송역의 열차번호 데이터를 수신하여 열차진로제어 테스트를 수행하는 것이다. 즉, 인접역의 열차출발 정보 등을 활용하여 인접역 모드 상태에서도 정상적으로 진로설정이 가능한지 테스트 하는 것이다.

아래 그림 6-38은 인접역 모드에서 오송역에서 출발한 열차(B3197)가 표시된 것을 확인할 수 있으며 열차 진입에 따라 신호기 4B에서 B1011T까지 진로가 정상적으로 설정되는 것을 확인하였다.

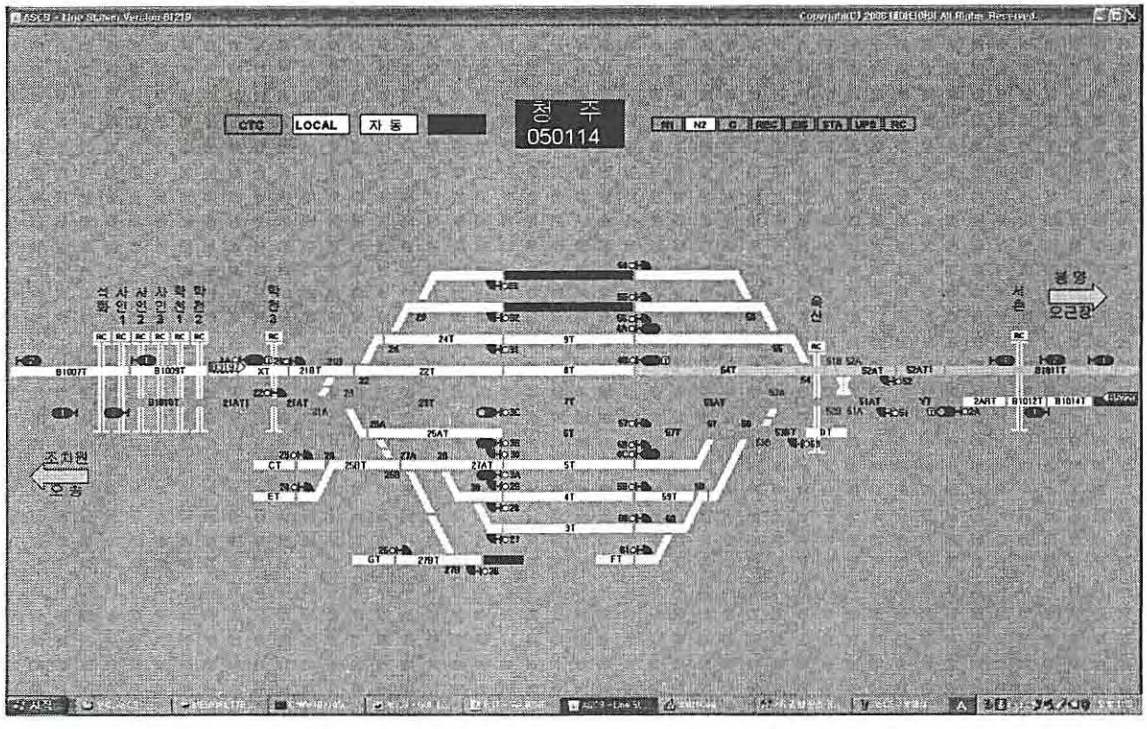


그림 6-38 역시스템 확장 단계의 열차진로제어 시험

3. 홈 리스케줄링 시험

가. 시험 목적

열차모니터링 서브시스템의 하나의 프로세스인 홈 리스케줄링 프로세스는 대형·고밀도운행 철도역의 열차운행 관리 최적화를 위한 프로세스로 인접역에서 들어오는 운행실적 및 현장 정보를 바탕으로 일정시간 내에 해당역에서 발생하는 경합을 검지하고 출발시간, 출발/도착 진로를 변경하여 열차지연을 최소화하는 변경스케줄을 생성하고 사용자의 승인 후 경로제어 서브시스템으로 전송하는 역할을 한다.

따라서 홈 리스케줄링의 현장시험의 목적은 홈 리스케줄링 프로세스가 ADM으로부터 열차위치표시정보, 역상태정보, TMS모드정보 등의 실제 현장데이터를 수신받아 정상 동작하는지와 현장 입환을 스케줄화 하였을 때 최적 변경스케줄을 작성하는지에 대한 테스트 항목을 도출하고 도출된 테스트 절차에 맞게 시스템을 테스트 하는 것이다.

나. 시험 환경

홈 리스케줄링 현장테스트의 시험 환경은 청주역 ASCS, 오송역 ASCS, 오근장역 ASCS가 온라인으로 실시간 구동되는 상태이며 양 방향 인접역의 정보를 모두 받을 수 있는 청주역 ASCS를 기준으로 현장시험 환경을 구축하였다. 그러므로 홈 리스케줄링 프로세스의 대상이 되는 열차는 오송역에서의 하행열차, 오근장역에서의 상행열차 그리고 청주역내의 입환 열차가 그 대상이 된다. HR대상시간은 1시간에서 최대 6시간까지 HR대상열차수는 10대에서 30대까지로 설정하였다.

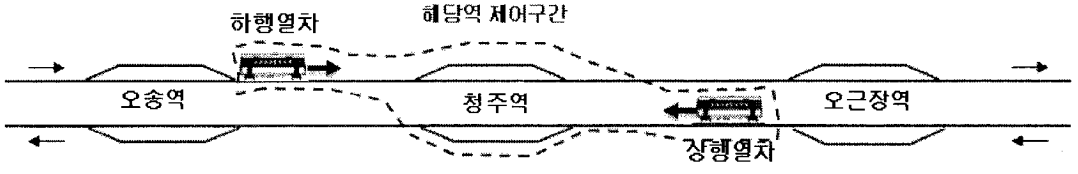


그림 6-39 홈 리스케줄링 시험대상 영역

다. 시험 절차 및 결과

(1) 인접역 열차 표시정보 활용 테스트

인접역 열차 표시정보 활용테스트는 크게 인접역 열차 표시정보 수신테스트와 HR대상열차 예측스케줄 업데이트로 나누어 수행하였다. 인접역 열차표시정보 수신테스트는 ASCS Data Field를 통하여 전송되는 인접역의 열차 표시정보를 HR에서 TMS의 ADM을 통하여 수신하는지를 테스트하는 것으로써 ASCS의 콘솔 화면 혹은 HR PARS로그파일을 통하여 수신여부를 확인할 수 있다.

HR대상열차 예측스케줄 업데이트는 수신 받은 열차 표시정보 중에서 해당역의 HR대상열차에 포함되어있는 열차의 스케줄을 인접역의 열차표시정보를 활용하여 새롭게 예측스케줄을 작성하는지를 테스트하는 것으로써 HR PROC로그파일을 통하여 수신여부를 확인할 수 있다.

(가) 인접역 열차 표시정보 수신 테스트

HR은 TMS의 ADM을 통하여 수신 받은 모든 CC 데이터를 2자리가 하나의 코드로 구성된 헥사코드형태로 출력하여 HR PARS로그파일로 남기고 있다. 이때, 0에서 시작하여 39번째 코드가 CC번호를 의미하며 열차표시정보는 '03'이다. 또한, 41번째~44번째 코드가 역번호를 의미하고 있다.

① 상행 열차 표시정보 수신

2009년 1월 28일 17시 36분에 인접역(오근장)에서 전송한 열차표시정보는 HR에서 수신 후에 아래와 같은 로그로 확인할 수 있다. 이때, 39번째 코드는 '03'으로 열차표시정보이며 41번~44번 코드는 '00 C4 27 54 4D'로 오근장역임을 확인하였다.

```
[2009-01-28 17:36:43] 41 53 43 53 01 01 00 00 01 01 00 00 49 80 13 21 00 00 0C 15 00 00 00
      00 00 00 00 00 00 00 00 64 49 80 19 32 01 01 00 03 00 00 C4 27 54 4D 53 32 54 50
      00 00 00 00 00 01 00 01 5F F9 49 80 19 01 00 63 42 33 30 35 31 56 32 30 30 39 30 31
      32 38 30 30 00 72 42 33 32 33 39 35 32 30 30 39 30 31 32 38 30 30
```

② 하행 열차 표시정보 수신

2009년 1월 28일 17시 39분에 인접역(오송)에서 전송한 열차표시정보는 HR에서 수신 후에 아래와 같은 로그로 확인할 수 있다. 이때, 39번째 코드는 '03'으로 열차표시정보이며 41번~44번 코드는 '00 C3 7C 54 4D'로 오송역임을 확인하였다.

```
[2009-01-28 17:39:19] 41 53 43 53 01 01 00 00 01 01 00 00 49 80 19 AC 00 00 00 02 00 00
      00 00 00 00 00 00 00 00 00 52 49 80 19 B1 01 01 00 03 00 00 C3 7C 54 4D 53 31 54
      50 00 00 00 00 00 01 00 01 5F F9 49 80 19 B6 01 12 42 33 31 37 38 35 32 30 30 39
      30 31 32 38 30 30
```

(나) HR대상열차 예측스케줄 업데이트 테스트

HR에서는 수신받은 열차 표시정보 중에서 해당역의 HR대상열차에 포함되어있는 열차의 스케줄을 인접역의 열차표시정보를 활용하여 새롭게 예측스케줄을 작성 하게 된다. 예측스케줄을 업데이트하는 로직은 수신 받은 열차표시정보(출발/통과)와 해당역 스케줄(정차/통과)을 바탕으로 HR대상열차의 TPS 패턴(Train Performance Simulator를 이용한 열차운행시분 산정값)을 계산한 후 메시지를 수신 받은 시간에 계산된 TPS패턴에 해당하는 열차이동시간을 더하여 새로운 예측스케줄을 작성하게 된다.

① 상행 열차 표시정보 수신

2009년 1월 28일 17시 36분에 인접역(오근장)에서 전송한 B3051열차의 열차표시정보를 바탕으로 B3051열차가 오근장역에서 정차하지 않고 통과(P)하는 열차이며 해당역(청주역)에서도 통과(P)하는 열차 즉, PP타입의 정차패턴을 가지고 있음을 확인한 후 메시지 수신시간 17시 36분 42초에 TPS패턴에 의한 열차이동시간 600초를 더하여 기존에 가지고 있는 17시 42분에 해당역(청주역) 도착이 아닌 17시 46분 22초에 도착하는 새로운 열차예측스케줄을 작성하고 이를 열차경합 검지 및 해소에 반영하게 된다.

```
[2009-01-28 17:36:43] [CC 수신정보] 역코드 : 050215 , 열차번호 : B3051_0, 열차속성 : 0x55
[2009-01-28 17:36:43] B3051_0 열차속성 : 상행, 출발, 통과
[2009-01-28 17:36:43] B3051_0 열차 예측시간을 다시 계산
[2009-01-28 17:36:43] DB로부터 050114RT2A_7T 진로의 예측시간 구함 , 통과시간 : 60 초, 정
      차시간 : 120 초
[2009-01-28 17:36:43] TPS 시간 : PP -> 600
[2009-01-28 17:36:43] TrainStateChange, TPS: 600, CURTIME: 173642
[2009-01-28 17:36:43] TPS 600, PassMove :60 , StopMove : 120, CurrentTime : 173642
[2009-01-28 17:36:43] TrainStateChange, dpt : 174622, arv : 174622 DB에 기록
[2009-01-28 17:36:43] CALL SP_PREDICSCH_UPDATE('B3051', 0, '20090128', '174622',
      '050114RT2A_7T', '174622', '050114RT3C_B1010T');
```

② 하행 열차 표시정보 수신

2009년 1월 28일 17시 36분에 인접역(오송)에서 전송한 B3178열차의 열차표시정보를 바탕으로 B3178열차가 오송역에서 정차하지 않고 통과(P)하는 열차이며 해당역(청주역)에서도 통과(P)하는 열차 즉, PP타입의 정차패턴을 가지고 있음을 확인한 후 메시지 수신시간 17시 39분 18초에 TPS패턴에 의한 열차이동시간 480초를 더하여 기존에 가지고 있는 18시 09분에 해당역(청주역) 도착이 아닌 17시 45분 18초에 도착하는 새로운 열차예측스케줄을 작성하고 이를 열차경합 검지 및 해소에 반영하게 된다.

```
[2009-01-28 17:39:19] [CC 수신정보] 역코드 : 050044 , 열차번호 : B3178_0, 열차속성 : 0x35
[2009-01-28 17:39:19] B3178_0 열차속성 : 하행, 출발, 통과
[2009-01-28 17:39:19] B3178_0 열차 예측시간을 다시 계산
[2009-01-28 17:39:19] DB로부터 050114RT1A_8T 진로의 예측시간 구함 , 통과시간 : 60 초, 정
    차시간 : 120 초
[2009-01-28 17:39:19] TPS 시간 : PP -> 480
[2009-01-28 17:39:19] TrainStateChange, TPS: 480, CURTIME: 173918
[2009-01-28 17:39:19] TPS 480, PassMove :60 , StopMove : 120, CurrentTime : 173918
[2009-01-28 17:39:19] TrainStateChange, dpt : 174518, arv : 174518 DB에 기록
[2009-01-28 17:39:19] CALL SP_PREDICSCH_UPDATE('B3178', 0, '20090128', '174518',
    '050114RT1A_8T', '174518', '050114RT4B_B1011T');
```

(2) TMS 모드정보 활용 테스트

TMS 모드정보 활용테스트는 크게 TMS 수동모드 테스트와 TMS 자동모드 테스트 두 가지로 구분하여 수행하였다. HR은 기본적으로 TMS가 자동모드일 때 열차경합을 검지하고 이를 해소하여 리스케줄링 하는 것으로 설계 되어있으므로 TMS의 모드 정보를 수신하여 TMS 모드가 수동일 경우에는 HR의 기본기능(HR 대상열차선정, 기본/예측스케줄 생성)은 정상적으로 동작하나 경합검지 및 해소는 발생하지 않는다. 이와 반대로 TMS 모드가 자동일 경우에는 HR의 기본기능 뿐만 아니라 경합 검지 및 해소 기능까지 정상적으로 동작하여야 한다. 본 테스트에서는 이를 HR PARS로그파일과 TMS-L/S상의 모드정보와 HR-GUI상의 모드정보의 일치 여부로 확인하였다.

(가) TMS 수동모드 테스트

HR은 TMS의 ADM을 통하여 수신 받은 모든 CC 데이터를 2자리가 하나의 코드로 구성된 헥사코드형태로 출력하여 HR PARS로그파일로 남기고 있다. 이때, 0에서 시작하여 28번째 코드가 TMS모드정보를 의미하여 수동일 경우에는 '01'이다. 또한, 아래 그림과 같이 TMS-L/S상의 모드정보와 HR-GUI상의 모드정보가 수동으로 일치하게 된다.

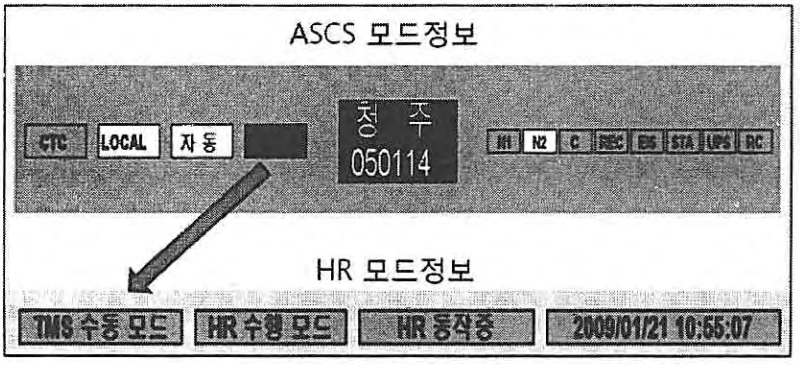


그림 6-40 TMS/HR 수동모드 일치 테스트

TMS 수동모드일 때는 기본기능 외의 경합검지 및 해소 기능은 처리되지 않음을 HR MMI상의 이벤트 로그를 통해 확인하였다. 아래 그림6-41과 같이 TMS 수동모드에서는 경합처리를 수행하지 않는다는 메시지를 사용자에게 알리며 사용자가 수동으로 경합해소를 요청하여도 경합처리는 수행되지 않는다는 알람이 발생되었다.

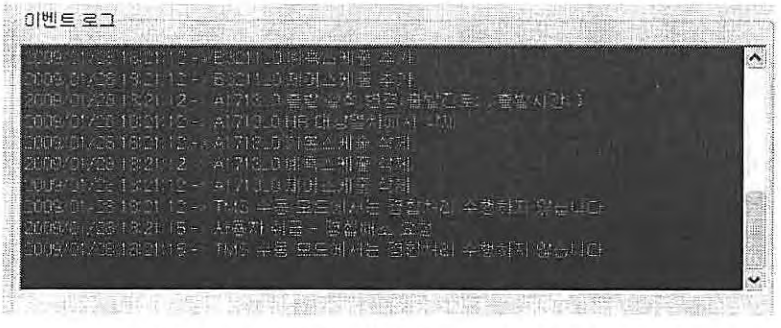


그림 6-41 TMS 수동모드 경합처리 로그

(나) TMS 자동모드 테스트

HR은 TMS의 ADM을 통하여 수신 받은 모든 CC 데이터를 2자리가 하나의 코드로 구성된 헥사코드 형태로 출력하여 HR PARS로그파일로 남기고 있다. 이때, 0에서 시작하여 28번째 코드가 TMS모드정보를 의미하여 자동일 경우에는 '00'이다. 또한, 아래 그림6-42와 같이 TMS-L/S상의 모드정보와 HR-GUI상의 모드정보가 자동으로 일치하게 된다.

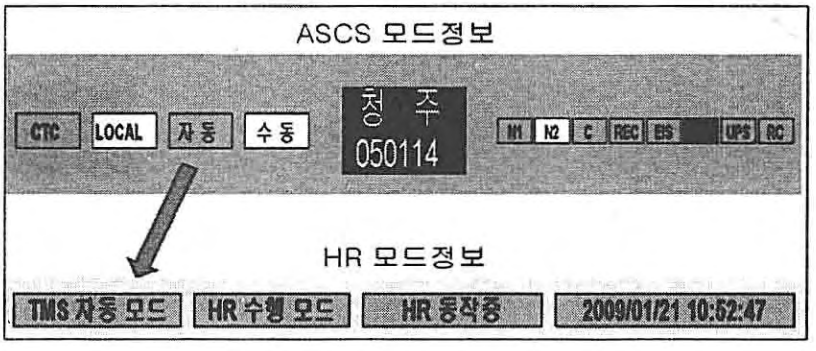


그림 6-42 TMS 자동모드 테스트

TMS 자동모드일 때는 기본기능 외의 경합검지 및 해소 기능 역시 정상 처리됨을 HR MMI상의 이벤트 로그를 통해 확인하였다. 아래 그림6-43과 같이 TMS 자동모드에서는 경합검지를 수행하였다는 메시지를 사용자에게 알리며 사용자가 수동으로 경합해소를 요청하면 경합해소를 수행하고 스케줄의 변경사항이 발생하지 않는 정보를 사용자에게 알람 하였다.

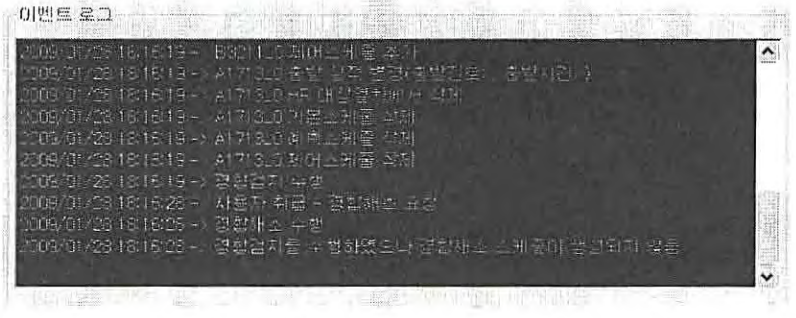


그림 6-43 TMS 자동모드 경합처리 로그

(3) 입환스케줄 경합 검지 및 해소 테스트

HR은 본선스케줄 뿐만 아니라 입환스케줄을 포함하여 경합을 검지하고 해소하는 기능을 가지고 있다. 하지만, 현장시험에서는 현재 역 운용원 및 수송원들이 따로 예측스케줄을 가지고 있지 않으므로 해당역에서 수행되어지는 역 입환 상황을 입환 스케줄화 하여 시험을 수행하였다. 입환스케줄 경합 검지 및 해소 테스트는 크게 입환이 하나의 입환스케줄로 구성된 경우, 입환이 하나 이상의 입환스케줄로 구성된 경우 두 가지로 시나리오를 구분하여 실시하였다.

본 연구에 사용하는 결합스케줄은 본선스케줄과 입환스케줄이 모두 포함되어있으므로 역 입환 상황을 본선스케줄과 같이 진로와 시간을 가지도록 구성하였다. 그렇기 때문에 현장에서 물리적으로 동일한 하나의 입환이 스케줄상으로는 하나 이상의 순서를 가진 입환스케줄로 표현이 된다. 예를 들어, 본선열차 A1804열차가 P3플랫폼에서 C를 거쳐서 Y1로 입환 하는 상황은 두 가지 입환 스케줄로 표현된다.

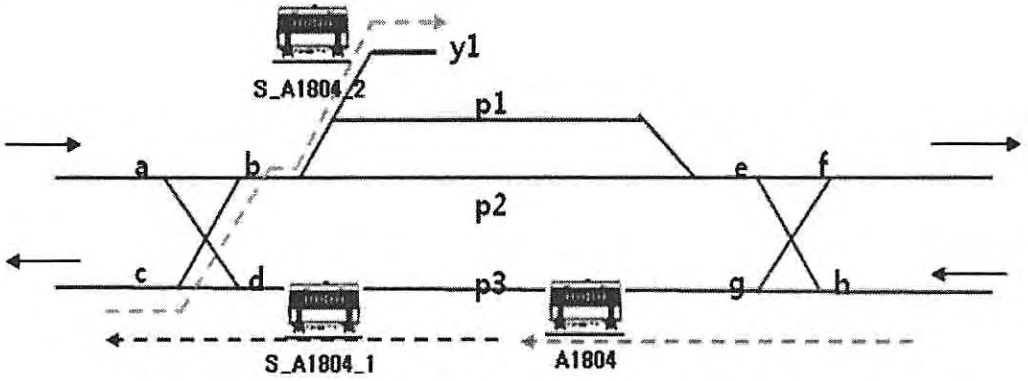


그림 6-44 입환 상황 예시

표 6-1 입환 상황 입환 스케줄화

열차번호	순서	계획 도착시간	계획도착진로	계획 출발시간	계획출발진로	비고
A1804	0	080000	h-g-p3			본선
S1804	1			081000	p3-d-c	입환
S1804	2			082000	c-b-y1	입환

(가) 입환이 입환스케줄 하나로 구성된 경우

오근장에서 출발하여 해당역(청주역)으로 2A_7T 진로를 사용하여 15:33:00에 도착하는 B3236열차가 있을 때 경합을 유발시키는 출발진로(59_YT) 및 출발시간(15:33:00)을 가지고 있는 하나의 입환스케줄로 구성된 입환열차 S0001_1을 삽입하였을 경우 HR에서는 경합을 검지하고 지연시간을 최소화하도록 S0001_1 입환을 계획에 따라 15:33:00에 수행하고 그 이후에 본선열차 B3236열차가 도착하도록 B3236열차를 15:34:40에 도착하도록 리스케줄을 생성하였다.

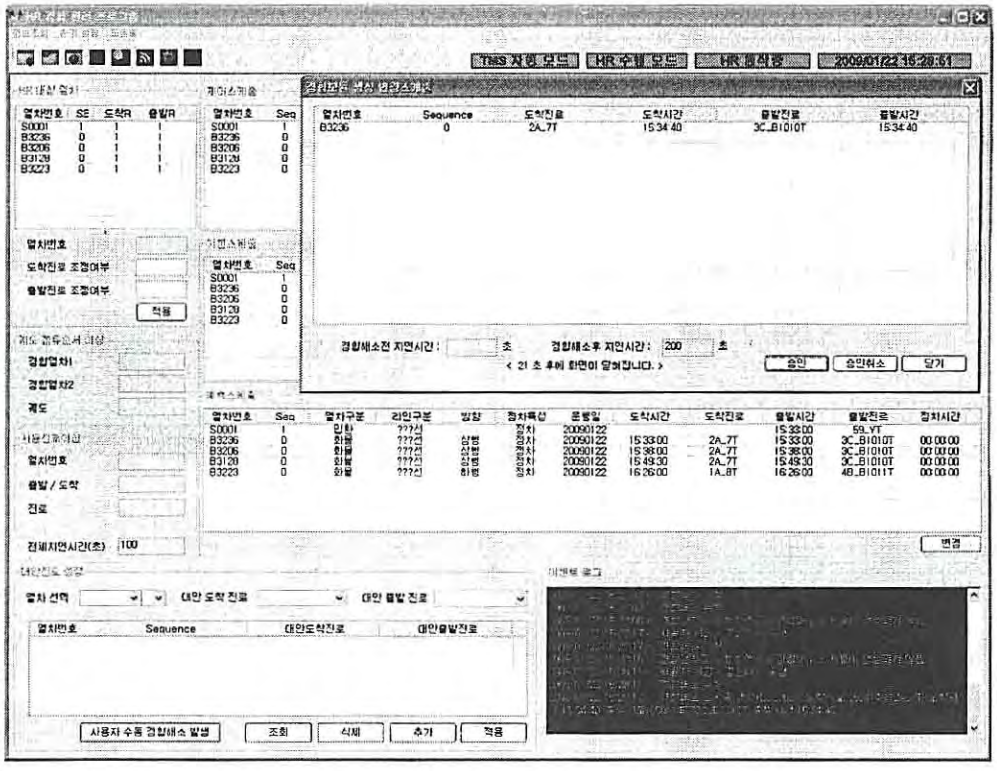


그림 6-45 입환스케줄 하나인 경우 리스케줄링

(나) 입환이 입환스케줄 하나 이상으로 구성된 경우

오근장에서 출발하여 해당역(청주역)으로 2A_7T 진로를 사용하여 15:33:00에 도착하는 B3236열차가 있을 때 경합을 유발시키는 출발진로 및 출발시간을 가지고 있는 두 개의 입환스케줄로 구성된 입환열차 S0001_1(출발진로 59_YT, 출발시간 15:33:00) S0001_2(출발진로 51_10T, 출발시간 15:35:00)를 삽입하였을 경우 HR에서는 경합을 검지하고 지연시간을 최소화하도록 본선열차 B3236은 계획에 따라

15:33:00에 수행하고 그 이후에 입환열차 S0001_1, S0001_2가 15:34:40, 15:36:40에 각각 출발하고 이후에 2A_7T 진로를 사용하여 15:38:00에 도착하게 본선열차 B3206열차가 15:38:20에 도착하도록 리스케줄을 생성하였다.

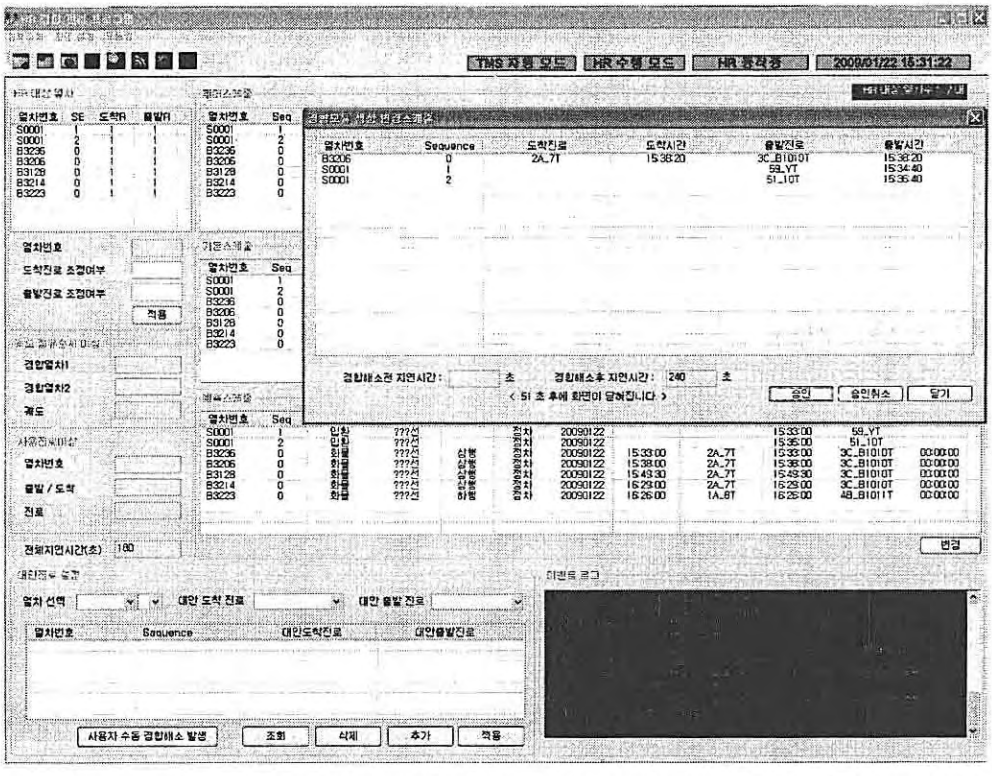


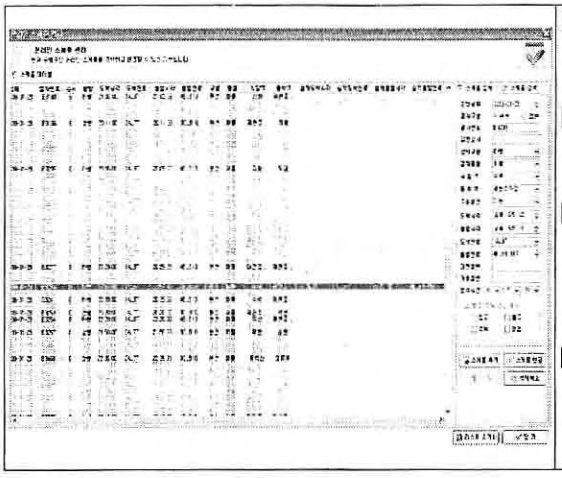
그림 6-46 입환스케줄 하나 이상인 경우 리스케줄링

(4) 열차 출/도착 실적정보 활용 테스트

열차 출/도착 실적정보 활용테스트는 HR대상열차 중에서 실제 열차가 출발하고 도착함으로써 출/도착 시간 및 진로가 실적을 기록하면 이를 HR에서는 인지하고 해당열차를 일정시간 후에 HR 대상열차에서 삭제하는지를 테스트하는 것으로써 HR PROC로그파일 및 MMI를 통하여 수신여부를 확인할 수 있다.

오송에서 출발하여 해당역(청주역)으로 1A_8T 진로를 사용하여 17:51:00에 도착하는 B3239열차가 계획보다 빠른 17:26:33에 도착하여 17:27:10에 출발하였을 때, 아래 그림과 같이 TMS-GUI를 통하여 B3239열차의 실적발생을 확인할 수 있으며 HR에서는 출발실적정보를 17:27:12에 B3239열차의 출발실적발생을 확인하고 해당 열차의 도착실적이 존재하는지 확인한 후에 일정시간(파라미터 : 1분)후에 HR대상

열차에서 B3239를 삭제함을 HRPROC로그파일 및 MMI를 통하여 확인하였다.



[2009-01-28 17:27:12] FindChangeSchedule 출발 실적정보 생김 B3239_0 Push

[2009-01-28 17:27:12] FindChangeSchedule AAAA B3239_0

[2009-01-28 17:27:12] FindChangeSchedule : B3239_0 도착실적이 있으므로 삭제등록 175100, Buf:052633

[2009-01-28 17:28:12] B3239_0 열차 : HR 대상 열차 / 기본스케줄 / 예측스케줄 / 제어스케줄에서 삭제

여 백

7장 결 론

1절 결론

본 연구에서는 현재 상시적으로 지역제어상태로 운영 중인 대형·고밀도운행 철도역(상시로컬역)의 당면문제를 근본적으로 해결하고, 우리나라 간선철도에서 사용하고 있는 중앙집중방식의 열차운행제어시스템 (CTC)의 기술적, 운영적 측면의 단점을 개선하기 위하여 차세대 열차운행제어체계로 자율분산방식 열차운행제어체계를 제시하고 자율분산 철도역 자동운행제어시스템 (ADTMS)을 개발하였다.

기존의 CTC체계에서는 중앙관제센터에서 원격으로 역의 열차트래픽을 자동 또는 수동으로 제어하는데 비하여 중앙(LMS)에서는 각 역의 최적 열차운행순서를 제시하고 각 역의 역제어시스템(ASCS)은 이 제시된 열차스케줄을 바탕으로 독자적으로 열차트래픽을 제어하는 방식을 취하여 중앙과 역현장의 역할을 분리하고 있다. 이 방식은 기존방식의 문제점인 대형역의 입환문제를 해결할 수 있을 뿐만 아니라 시스템의 개발, 설치 및 유지보수 측면에서 단계적인 증설, 온라인 유지보수 기능을 가질 수 있는 장점이 있다.

ADTMS는 크게 중앙시스템인 노선관리시스템과 역제어시스템으로 구성되며, 노선관리시스템은 라인모니터링, 라인경합관리 및 라인정보 서브시스템을, 역제어시스템은 실시간 열차추적, 경로제어, 열차모니터링 및 수송관리 서브시스템을 가지도록 설계되었다. 노선관리시스템과 역제어시스템은 동등한 권한을 가지도록 설계되어 있어 어느 시스템의 고장 경우에도 역제어시스템은 역내 정보를 이용하여 최소한의 운행제어기능을 수행할 수 있다.

개발된 시스템의 시험을 위하여 주관연구기관 내에 종합연계시험센터를 구축하고 시뮬레이터를 이용하여 기능시험 및 연계시험을 마쳤으며, 현장에서의 검증을 위하여 충북선 오송역-청주역-오금장역에 역제어시스템을 설치하여 현장시험을 통하여 문제점을 보완하였다.

향후 이 개발된 시스템은 다음과 같은 목적으로 활용될 것으로 기대된다.

- 기존 CTC체계하에서 대형·고밀도 운행이 필요한 역의 자동운행제어시스템
- 열차트래픽이 낮은 복수 개의 인접역을 연계하여 하나의 ASCS로 제어할 수 있는 소지역제어시스템

- 새로 건설되는 신선을 위한 완전한 ADTMS 시스템

2절 향후 연구계획

본 연구결과 개발된 시스템은 향후 추후 연구를 통하여 실용화할 예정이다. 실용화에 예상되는 주요 사항은 다음과 같다.

실용화연구의 연구기간은 최소 1년에서 최장 3년까지 점진적으로 적용범위를 확대하며 매년 연구결과를 도출할 수 있도록 하는 것이 바람직하다.

실용화를 위한 주요 연구내용은 다음과 같다.

- 개발된 시스템의 성능 및 안전 인증 획득 연구
  - 성능 및 안전인증을 위한 사양 및 절차 수립
  - 인증시험 사양 및 절차에 따른 종합연계시험장치 개선
  - 인증절차 수행
- 역제어시스템의 기능통합 및 비용 축소
  - 현재 3개로 분리되어 있는 서브시스템을 하나로 통합
  - 최적화모듈의 유지보수방안 개선
- 역제어시스템의 제어정밀도 향상
  - 경합검지를 위한 열차위치 파악 정밀도 개선
- 열차운용원을 위한 GUI 개선
  - 최단시간에 효율적으로 열차입환스케줄을 입력할 수 있도록 개선

예상되는 기대효과는 다음과 같다.

- 국내 대형역은 대부분 상시현장제어로 운영되고 있으며, 역운행제어시스템은 현재 운영중인 CTC와 연계운용이 가능한 시스템으로 안정화 생산이 되면 국내시장 점유는 가능할 것으로 판단하고 있음
- 전국적인 대형시스템의 상용화 이전이라도 소수의 역을 동시에 제어하는 소규모 열차운행제어시스템으로 우선 상품화함으로써 응용분야를 점진적으로 확

대 가능

- 국외시장은 주로 철도시스템의 진출과 동반되는 경향이 강하고, 안전시스템의 특성으로 인하여 국내 운용실적이 중요함. 따라서 국내 운용 후 철도시스템의 진출과 동반하여 국외에 진출하는 전략을 가지고 있음
- 연구개발을 통하여 구축한 「종합연계시험장치」의 개선과 인증체계 수립을 통하여 열차운행제어시스템의 인증업무에 활용

참고문헌

1. 건설교통부. '대형·고밀도운행 철도역 자동운행제어시스템 개발(1차년도 중간보고서), 한국철도기술연구원, 2004. 4.
2. 건설교통부. '대형·고밀도운행 철도역 자동운행제어시스템 개발(2차년도 중간보고서), 한국철도기술연구원, 2005. 5.
3. 건설교통부. '대형·고밀도운행 철도역 자동운행제어시스템 개발(3차년도 중간보고서), 한국철도기술연구원, 2006. 5.
4. 건설교통부. '대형·고밀도운행 철도역 자동운행제어시스템 개발(4차년도 중간보고서), 한국철도기술연구원, 2007. 5.
5. 홍순흠, '철도운행제어시스템 기술의 최근 동향 및 전망', 대한전자공학회 전자공학회지, 2007. 11.
6. 김영훈, '자율분산 역제어시스템 모의시험 연구' 한국철도학회 추계학술대회논문집, 2008.11.
7. A. D'Ariano Conflict resolution and train speed co-ordination for solving timetable perturbations, RailDelft 2005
8. E. Lamma, P. Mello & M. Milano, "A distributed constraint-based scheduler", Artificial Intelligence in Engineering, 91~105P, 1997
9. Eiji Nishijima, "On-line Testing for Application Software of Widely Distributed System", IEEE, 1996
10. ELLO A. G. WEITS, "Simulation of Railway Traffic Control", Int. Trans. Opl Res. Vol. 5, No. 6, pp461-469, 1998
11. F. Kera et al., "Assurance System Technologies for Large Scale Transport Operation Control System" , Technical Report of IEICE, 1996.
12. F. Kitahara et al., "Phased-In Construction Method of ATOS", Proc. Of ISADS99, 1999.
13. F. Kitahara, "Autonomous Decentralized Traffic Management System", J.IEE Japan, 1999.
14. F.Kitahara et al., "Distributed Management for Software Maintenance in a Wide-Area Railway System", Proc. Of ISADS97, 1997.

15. F.Kitahara et al., “Widely-Distributed Train-Traffic computer Control System and Its Step-by-Step Construction”, Proc. Of ISADS95, 1995.
16. Fumio Kitahara et al., "Widely-Distributed Train-Traffic Computer Control System and Its Step-by-Step Construction, IEEE. 1995
17. Hirokazu Ihara, "Highly reliable loop computer network system based on autonomous decentralized concept", IEEE, 1982, 187-194
18. Hirokazu Ihara, "More dependable systems in computing - Autonomous Decentralized Systems-", DCIA 1998.
19. Hiroshi Wataya et al., "The Cooperating Autonomous Decentralized System Architecture", IEEE. 1995
20. Hiroshi Yamanomto, "On-line Test Technique Based on Autonomous Decentralized System", IEEE, 1993
21. Hitachi, NeXUS/Dlink Guide & Reference software manual
22. IRJ(International railway journal), 2003. 12
23. Isao JAJI, "A gateway filtering technique to maximize the transactions in heterogeneous systems", IEICE vol.E84-B, NO10 OCT 2001
24. Joseph Schmuller, UML 객체지향설계 제3판, 정보문화사 2004
25. K. Mori, "Autonomous Decentralized System", 電子情報通信學會, 2001
26. Katsumi Kawano, "Autonmous Decentralized System Test Technique", IEEE, 1989
27. Kazue Kera, "Adaptive step-by-step construction technique achieving high assurance for Tokyo Metropolitan railway network", IEEE, 2001
28. Kazuo KERA et al., “Assurance Technology for Growing System and Its Application to Tokyo Metropolitan Railway Network”, IEICE TRANS. INF. & SYST, OCTOBER 2001.
29. Kazuo Kera, "Assurance technology for Tokyo Metropolitan railway network", IEEE 2002
30. Kinji Mori, “Autonomous Decentralized Systems: Concept, Data Field Architecture and Future Trends” , Proc. Of ISADS93, 1993.
31. Kinji Mori, "Expandable and Fault Tolerant Computers and Communications

Systems-Autonomous Decentralized Systems", IEEE. 1999

32. L.A. Wolsey, Integer Programming, John Wiley and Sons, INC. 1999
33. Masayuki Orimo et al, "Autonomous Decentralized System Supporting for Fault-Tolerance and On-line Maintenance of Communication Software", IEEE. 1995
34. Masayuki Orimo, "Autonomous Decentralized File System and Its Application", IEEE, 1992
35. Rational Rose, Using Rose Data Modeler, Version:2001A.04.00
36. S. Shoji et al., "New trends of train control and management systems with real-time and non real-time properties", Proc. Of ISADS97, 1997.
37. Takashi Kawakami, "Concept of Friendly autonomous decentralized system for next generation train traffic control", IEEE, 1993
38. Takashi Kawakami, "The new system architecture for next-generation Shinkansen based on the friendly autonomous decentralized system concept", IEEE, 1995
39. Takeuki Aizono "An Object Model of Autonomous Control Processor and the Message Selection Method"
40. Yoshio Ishida, "Strategic railway systems using ADS technologies in Japan", IEEE, 2003
41. U. Brannlund, P. O. Lindberg, A. Nou, and J. E. Nilsson. "Rail-way timetabling using lagrangian relaxation". Transportation Science, 32(4):358-369, 1998.
42. A. Caprara, M. Fischetti, and P. Toth. "Modeling and solving the train timetabling problem". Operations Research, 50(5):851-861, 2002.
43. Malachy Carey and David Lockwood. "A model, algorithms and strategy for train pathing". The Journal of the Operational Research Society, 46(8):988-1005, 1995.
44. A. Higgins, E. Kozan, and L. Ferreira. "Optimal scheduling of trains of a single line track", Transportation Research Part B: Methodological, 30(2):147-161, 1996.

45. A. Higgins, E. Kozan, and L. Ferreira. "Heuristic techniques for single line train scheduling", Journal of Heuristics, 3(1):43-62, 1997.
46. Carey M. "A model and strategy for train pathing with choice of lines, platforms, and routes", Transportation research. Part B : methodological, 28(5) · 333 - 353, 1994. 39
47. Maura Mazzarello and Ennio Ottaviania. "A traffic management system for real-time traffic optimization in railways", Transportation Research Part B: Methodological, 41(2):246-274, 2007.
48. Szpigel.B. "Optimal train scheduling on a single track railway". Operations Research, p.343-352, 1973.
49. Xuesong Zhoua and Ming Zhong. "Single-track train timetabling with guaranteed optimality". Transportation Research Part B: Methodological, 41(3):320-341, 2007

주 의 사 항

1. 본 보고서는 국토해양부가 출연하고 한국건설교통기술평가원에서 위탁 시행 한 건설교통기술연구개발사업의 최종연구보고서 입니다.
2. 본 보고서의 내용을 발표할 때에는 반드시 국토해양부가 출연하고 한국건설교통기술평가원에서 위탁시행 한 건설교통기술연구개발사업임을 밝혀야 합니다.
3. 국가보안 차원에서 필요하다고 인정되는 내용은 대외적으로 발표 및 공개하여서는 아니됩니다.

본 보고서와 관련하여 문의를 원하시는 분은 아래의 문의처로 연락을 주시기 바랍니다.

■ 문의처 : 한국건설교통기술평가원 TEL 031)381-6311
한국철도기술연구원 TEL 031)460-5481
(주)대아티아이 TEL 032)680-0882

R&D / 03교통핵심A02

대형·고밀도운행 철도역 자동운행제어시스템 개발 연구보고서

■ 발행일 / 2008. 12 . 31 .
■ 발행처 / 한국철도기술연구원
경기도 의왕시 월암동 360-1번지
TEL : 031-460-5000(대)
■ 인쇄처 / 성균사