

대형·고밀도 철도역 자동운행제어시스템의 핵심서브시스템 설계와 개발

Designing & Development of Prototype of Core Subsystem for High Density & Large Scale Station Traffic Management System

안 진† · 이영수* · 김은희* · 하승태* · 홍순흠** · 김영훈***

Jin Ahn · Young-Su Lee · Eun-Hee Kim · Sung-Tae Ha · Soon-Heum Hong · Young-Hoon Kim

Abstract

In the current train operation system, large scaled stations like Cheongnyangni and Yongsan are not remote controlled from the central operator but manually by local operator. It is because there are so many entering and exit of trains between depot and mainline which are operating through the large scaled stations and it is very effective to traffic flow on the mainline. Therefore, Kyong Bong Technology Co., Ltd. (KBTech) and Korea Railway Research Institute (KRRRI) are continuing the research and development to import 'Autonomous Decentralized System (ADS)' as part of a new train operation control system which is a national policy project of MOCT (Ministry Of Construction & Transportation). The concept of the 'Autonomous Decentralized System (ADS)' has born in 1970's in Japan by the influences of molecular biology. The special features of ADS are being a system that able to do the online expansion, online maintenance and fault tolerance. This thesis introduces the concept of technology of ADS and also describes the system designing & development to apply with large scaled stations.

Keywords : Autonomous Decentralized System(자율 분산 시스템), High density & Large Scale Station(대형 고밀도 역), Train Management System(열차 운행 관리 시스템), Online Expansion(온라인 확장)

1. 서 론

현재 대형·고밀도 철도역의 열차 운행 제어는 중앙집중식 열차운행제어 시스템에서 주 본선에 대한 제어만 가능한 수준이나 이마저도 수시로 변동되는 역의 운행 상황을 CTC 서버에서 완벽하게 대처가 불가능하여 역을 운행하는 열차의 모니터링 기능만 수행하는 상시 로컬역으로 운영되고 있다. 이와 같은 상황이 발생하는 것은 대형역일수록 본선의 열차 흐름에 영향을 주는 열차의 입출고가 빈번하고 또한 여러 노선이 병행되어 운행되고 있기 때문이다.

자율분산형 열차운행제어체계는 이러한 상시로컬로 운영되는 대형역에 기본적으로는 열차 스케줄에 의한 자동제어 기능을 제공하고, 부가적으로 역 운영자에게 필요한 구내 입합 작업에 대한 스케줄 화, 홈 리스케줄링 등의 기능을

제공하여 이러한 현업의 문제점을 해결하는 데에 기여할 수 있다. 여기에서는 자율 역 제어 시스템의 핵심 서브시스템인 실시간 열차추적 서브시스템과 경로제어 서브시스템, 열차 모니터링 서브시스템의 설계와 테스트 시나리오를 살펴본다.

2. 자율분산 시스템 기술의 배경

2.1 배경

컴퓨팅 기술과 네트워크 기술의 발전으로 시스템은 점점 대형화, 복잡화되고 있다. 기존에는 Top-Down 방식으로 시스템 전체의 목적과 기능을 정한 다음 구축하는 접근방식을 취하였다. 이러한 접근방법으로는 규모나 구성, 기능을 유연하게 변경하고 확장하기 어렵다는 문제가 있다. 자율분산 시스템은 이러한 배경 하에서 사회 경제적인 상황변화에 적응할 수 있는 시스템 아키텍처로서 일본에서 1970년대 후반부터 연구가 진행되어 1990년대 후반에 시스템에 적용되어 개발된 개념이다.

† 책임저자 : 회원, 경봉기술(주), 설계/R&D그룹
E-mail : jinahn@kbtech.co.kr
TEL : (032)680-0873 FAX : (032)680-0800

* 경봉기술(주)

** 회원, 철도기술연구원, 책임연구원

*** 회원, 철도기술연구원, 선임연구원

2.2 자율분산시스템의 특징 및 지향점

자율분산은 온라인 확장성, 온라인 보수성, 플트 톨러런스성을 그 특징으로 한다. 온라인 확장성은 시스템을 가동시킨 채 변경하거나 확장할 수 있는 것을 말한다. 온라인 보수성은 시스템을 가동시킨 채로 시스템을 보수할 수 있는 것을 말한다. 플트 톨러런스는 시스템이 부분적인 장애가 생길지라도 그 장애의 파급을 저지할 수 있어 시스템 전체의 가동 정지를 피할 수 있는 것을 말한다.

2.3 시스템의 관점

이러한 자율분산시스템에서는 시스템을 바라보는 관점이 중앙집중식과 다른데 그것은 시스템은 서브시스템의 집합으로 규정하는 것이다. 전체 시스템의 목적, 기능을 먼저 규정하는 것이 아니라 서브시스템의 목적, 기능을 먼저 규정한 후에 그의 집합으로 바라보는 것이다. 또 하나는 시스템은 항상 기능이 비가동중인 서브시스템을 포함한다는 것이다. 기존과 달리 장애·구축중·보수중인 서브시스템이 있는 것을 정상적인 상태로 바라보는 것이다. 다음은 서브시스템의 비가동(가동중단) 상태가 전체 시스템에 영향을 주지 않는다는 것이다. 이를 위해서 자율분산시스템은 다른 서브시스템의 기능이 비가동 상태가 되더라도 스스로 제어할 수 있는 자율 제어성을 갖고, 다른 서브시스템의 기능이 비가동 상태가 되더라도 스스로 나머지 서브시스템과 협조할 수 있는 자율 협조성을 갖는다.

2.4 서브시스템의 조건

이러한 관점에서 자율분산시스템이 구성되기 위한 몇 가지 요건이 있는데 그것은 균질한 구조를 가지고 있어 어떠

한 서브시스템도 접속할 수 있도록 동일한 입출력 구조를 가져야 한다. 또 하나는 기능이 평등해야 한다. 다른 서브시스템으로 명령을 내리거나 명령을 받아 움직이는 것이 아니다. 다음은 국소적인 정보에 기초해야 한다. 시스템은 장애 보수중인 서브시스템을 포함하기 때문에 전체 시스템은 항상 변화하므로 전체적인 정보에 기초할 수가 없다.

3. 자율분산 시스템 기술을 적용한 자율분산 역 제어 시스템 설계

이 절에서는 시스템 소프트웨어인 자율분산관리자와 응용 소프트웨어인 프로세스간의 인터페이스 구조설계를 살펴보고, 응용 프로세스의 구성을 살펴본다.

3.1 자율분산관리자와 응용 프로세스의 분리된 아키텍처 설계

자율분산 시스템의 구성원인 각각의 서브시스템은 개별로 하나의 자율분산관리자를 갖는다. 자율분산관리자는 시스템 소프트웨어로 '철도역 열차운행제어'라는 비즈니스 영역에서 요구하는 열차 추적, 경로제어와 같은 비즈니스 로직을 담지 않는다. 이러한 비즈니스 로직은 응용 소프트웨어인 각각의 프로세스들이 담당한다. 자율분산관리자는 프로세스의 관리 기능과 통신 기능을 담당한다.

응용 프로세스는 시스템이 가동 중에도 기능이 변경되거나 메시지가 변경될 수 있다. 이러한 응용 프로세스가 변경되면 자율분산관리자는 기존에 실행중인 프로세스들을 종료하고 새로운 프로세스를 재 로드한다.

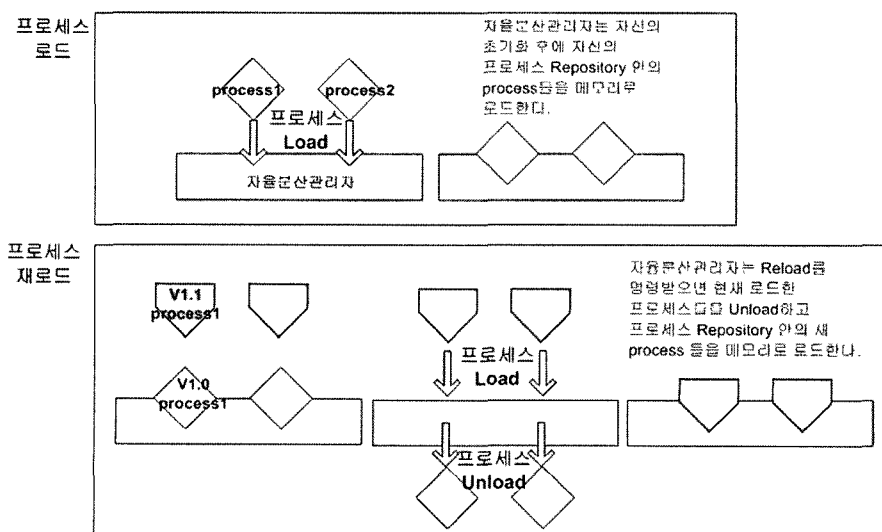


Fig. 1. Application Software Reload by ACP (Autonomous Control Process)

Fig. 1과 같이 자율분산관리자가 맨먼저 기동된 다음에, 자율분산관리자는 응용 프로세스가 보관된 Repository에서 프로세스를 기동시킨다. 응용 프로세스를 변경해야 하는 경우가 발생했을 때에 유지보수자는 자율분산관리자가 프로세스 재로드하도록 명령한다. 자율분산관리자는 먼저 로드했던 응용 프로세스를 내리고 새로운 버전의 응용 프로세스를 로드한다.

이렇게 응용 프로세스의 버전업 과정에서도 자율분산관리자는 계속 기동되어 해당 노드의 통신과 관리 기능을 지속적으로 수행하게 된다.

3.2 자율분산관리자와 응용 프로세스간의 인터페이스 구성 요소

자율분산관리자와 응용 프로세스간의 인터페이스의 구성요소를 살펴보자.

다음은 Fig. 2의 각각의 항목에 대한 설명이다.

- **자율분산관리자** : 각각의 서브시스템의 기동은 먼저 자율분산관리자의 기동으로부터 출발한다. 자율분산관리자는 자신의 초기화가 완료되면 프로세스 Repository안의 프로세스 파일들을 자신의 메모리 영역으로 로드한다.
- **process1, process2(응용 프로세스)** : 비즈니스 로직을 구현한 응용 프로세스 클래스의 집합이다. 비즈니스 로직 구현 클래스들 뿐만 아니라, 해당 프로세스 정보, 메시지 정보 등을 담고 있다.
- **interface(자율분산관리자의 인터페이스)** : 자율분산관리자가 응용 프로세스에게 기능을 제공하기 위한 추상 인터페이스의 집합이다. 응용 프로세스는 이 추상 인터페이스를 상속받아 구현된다. 프로세스 인터페이스, 메시지 인터페이스 등을 제공하여 응용 프로세스에서 자

율분산관리자의 기능을 사용하도록 제공한다.

- **process 정보** : 해당 응용 프로세스의 정보를 담고 있다. 자율분산관리자는 이 정보를 통해서 프로세스의 invoke 클래스를 판단한다.
- **message 정보** : 해당 응용 프로세스가 송수신하는 메시지의 정보를 담고 있다. 메시지 별로 내용 코드(Contents Code : CC), 메시지 설명, 클래스 정보, 송수신 여부를 담고 있다. 모든 송수신되는 메시지들은 자율분산관리자의 interface에 정의된 message 인터페이스를 상속하여 구현된다. 자율분산관리자는 이 정보를 통해서 응용 프로세스가 송신하려는 메시지에 내용코드를 부여하고 송신한다. 수신시에도 이 정보를 참조하여 내용코드에 따라 해당 메시지 객체를 생성하여 응용 프로세스에 넘겨준다.
- **config 정보** : 자율분산관리자는 이 정보를 통해 해당 시스템의 정보를 얻어온다. 시스템의 설명, 노드번호, IP, Port 등을 담고 있다.
- **node 정보** : 협력할 노드 정보들을 담고 있다. 협력할 노드 간에 이 정보는 동일해야 한다.
- **ccode 정보** : 메시지를 송수신하는 데에 필요한 정보를 담고 있다. 송수신방법은 브로드캐스트인지 또는 P2P인지, 브로드캐스트는 어느 어느 데이터 필드의 어느 멀티캐스트 그룹으로 송신할 것인지, P2P의 송신노드 등의 정보를 담는다.

3.3 자율분산관리자와 응용 프로세스간의 메시지 송수신 메커니즘

3.3.1 응용 프로세스간의 직접적인 메시지 송수신 배제 서브시스템의 자율분산관리자에 등록된 두 개 이상의 프

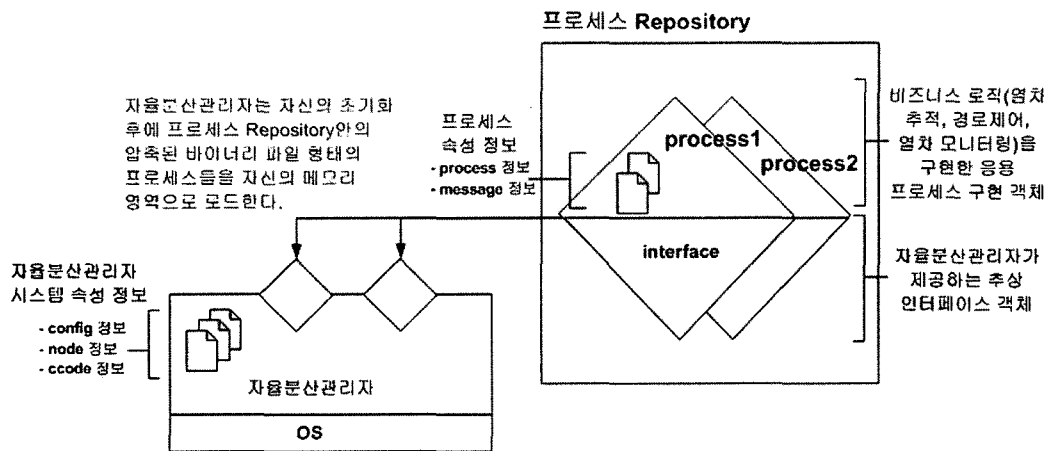
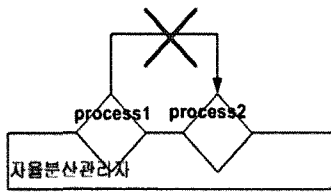


Fig. 2. Configuration Items of Subsystem's Software



자율분산관리자를 배제하고 응용 애플리케이션간 직접적인 메시지, 송수신, 호출등 연결 관계를 배제함

Fig. 3. Exclusion of Interface between Software Modules in one Subsystem

로세스 간에 메시지를 송수신 하는 경우에도 반드시 자율분산관리자를 통해서 메시지를 전달하도록 한다. 응용 소프트웨어간의 직접적인 연결 관계는 서로간의 의존성을 갖게 되기 때문이다.

Fig. 3과 같이 응용 프로세스끼리 자율분산관리자를 배제하고 통신을 하여 서로 응용 모듈을 실행하도록 명령을 주고 받는 인터페이스는 존재하지 않는다.

3.3.2 내부 데이터필드와 외부 데이터필드로의 동시 송출 지원

자율분산관리자의 응용 프로세스에서 출력되는 메시지가 서브시스템 내부의 프로세스와 외부 시스템의 다른 자율분산관리자의 프로세스에 동일하게 전달되어야 하는 경우 자율분산관리자는 내부 데이터 필드로의 전송과 외부 데이터 필드로의 전송을 모두 수행한다. 자율분산 관리자는 동일 서브시스템 내의 프로세스로는 내부 데이터필드를 통해서 전달하고, 외부의 자율분산관리자에 속한 응용 프로세스로는 외부 데이터필드를 통해서 데이터를 전달한다. 이러한 메시지의 송수신은 프로세스 속성 정보와 자율분산관리자의 시스템 속성 정보에 근거하여 수행된다.

Fig. 4와 같이 자율분산관리자는 process2가 송신한 메시

지를 해당 서브시스템 내부의 데이터필드와 서브시스템간의 데이터필드 양쪽에 송신한다. 그래서 같은 서브시스템 내에 있는 process1과 다른 서브시스템에 있는 process3, process4에 모두 자율분산관리자를 통해서 전달된다.

3.4 응용 프로세스인 핵심 서브시스템 설계

앞서 말한 서브시스템은 하나의 역 시스템으로써 자율역 제어 시스템(Autonomous Station Control system; ASCS)이라고 명명하였다.

이 각각의 ASCS가 모여 선구 시스템을 이루고 이러한 선구 시스템들이 모여 전체 시스템을 이룬다. 여기서는 하나의 역 시스템을 구성하는 서브시스템 중에서 핵심 서브시스템의 설계에 대해서 살펴본다.

Fig. 5와 같이 핵심 서브시스템은 기능별로 열차 추적을 담당하는 RTPS(Realtime Train Positioning System), 경로 제어를 담당하는 RCS(Route Control System), 열차 모니터링을 담당하는 TMS(Train Monitoring System)로 나뉘어져 있다.

RTPS 서브시스템에는 LDTS(Local Data Transmission System) 와의 인터페이스를 담당하는 표시정보처리 프로세스, 열차 추적을 담당하는 열차번호관리 프로세스로 구성된다.

RCS는 열차의 리스케줄링을 담당하는 홈 리스케줄링 프로세스와 그 리스케줄링 결과에 따라서 경로제어를 수행하는 경로제어 프로세스로 구성된다 현재 수행중인 3차년도는 홈 리스케줄링 로직을 구현하는 단계로써 구성도에 표시하지 않았다. TMS는 화면 표시 프로세스와 현장 제어 프로세스로 구성된다.

그리고 이러한 응용 소프트웨어는 모두 자율분산관리자라는 프레임워크 위에서 구동되어 자율분산관리자를 통해서 메시지 송수신을 한다. 따라서 모든 응용 소프트웨어는 자율분산관리자라는 동일한 입출력구조를 갖는다. 자율분산관리자는 프레임워크로써 시스템 소프트웨어로, 비즈니스

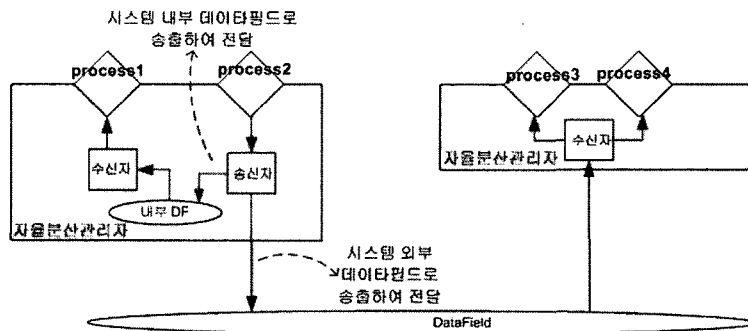


Fig. 4. Separation between Datafield of inner Subsystem and Datafield of inter Subsystem by ACP

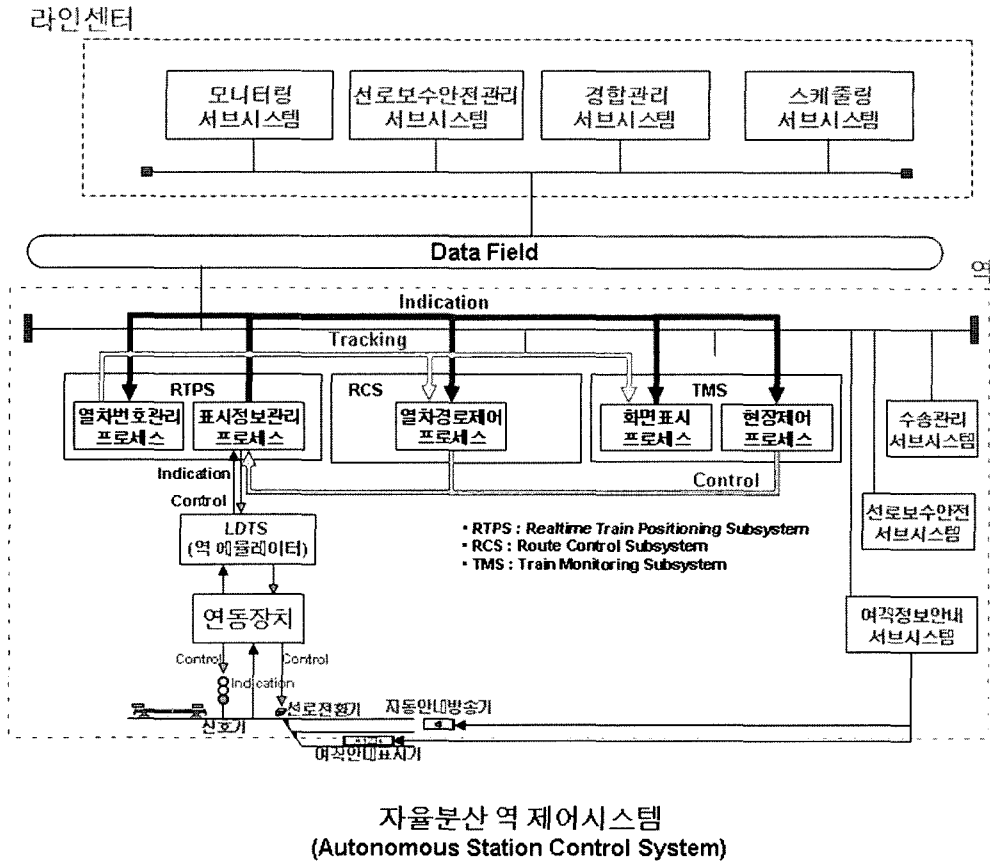


Fig. 5. Data Flow of Core Subsystem of Station

자율분산역 제어시스템
 (Autonomous Station Control System)

스 로직을 구현한 프로세스들은 응용 소프트웨어로 구분해 볼 수 있다. 이 응용 소프트웨어는 자율분산관리자 위에서 구동되고 송수신할 수 있도록 자율분산관리자가 제공하는 추상 인터페이스들을 상속받아 구현한다. 또한 자율분산관리자의 시스템 속성 정보에 자신의 프로세스 정보, 노드 정보, 메시지 정보 등을 등록하여 자율분산관리자가 등록된 프로세스 정보에 의해서 구동하고 등록된 메시지들을 송수신하도록 한다.

핵심 서브시스템간의 주요 데이터 흐름이 Fig. 5에 나타내었다. 현장설비에 대한 표시 및 제어 정보는 모두 RTPS의 표시정보관리 프로세스를 거치도록 구현되며 시스템간의 데이터 송수신은 모두 자율분산관리자를 경유하여 이루어진다.

송수신 메시지는 주기적으로 송신되는 주기성 메시지는 브로드캐스트 방식으로, 이벤트성으로 송신되는 메시지는 P2P(Point-to-Point) 방식으로 송신된다. 이러한 메시지별 통신방법의 결정과 송신 노드의 결정은 모두 자율분산관리자의 시스템 속성 정보에 등록된 대로 수행된다.

4. 시스템 테스트 시나리오

여기에서는 자율역 제어 시스템이 온라인 속성들, 즉 온라인 확장, 온라인 유지보수, 온라인 테스트가 가능한 가를 검증하는 테스트 시나리오를 설명한다.

4.1 온라인 속성 구현

온라인 확장, 온라인 테스트, 온라인 유지보수성은 각각의 응용애플리케이션이 자율분산이라는 동일한 인터페이스를 가진다는 점과 온라인/테스트 데이터필드의 구분을 통해서 달성된다.

분산시스템으로써 이 시스템은 각각의 시스템은 하나의 동일한 단위를 이루어 개별 응용 소프트웨어간의 인터페이스 없이 자율분산관리자에 의해서 메시지 송수신을 함으로써 상호 운영된다. 이들 응용 소프트웨어들은 다른 어플리케이션에 대해 독립적인 구조를 가지기 때문에 개별적으로 변경되어 Version될 수 있다. 이전 버전의 응용 소프트웨어는 자율분산관리자에 의해서 Unload되고 새로운 버전의 응용 소프트웨어가 Reload 된다. 이러한 작업들은 원격시스템

에 의해서 수행을 명령할 수 있다.

4.1.1 온라인상 새로운 애플리케이션 프로세스의 추가 시나리오

온라인 확장의 기본 전제로써의 논리는 전체 시스템의 구조는 시스템 구축 초기에 결정되지 않는다는 것이다. 각각의 서브시스템이 결정되면 그 서브시스템들의 집합으로써 시스템이 결정되는 것이다.

Fig. 6에서는 어떠한 역이 처음에는 홈 리스케줄링 기능이 필요하지 않아서 설치되지 않고 운영중인 상태이다가 점차 운행 패턴이 복잡해져서 홈 리스케줄링 기능이 필요하게 된 상황이라고 가정한다. 현재 이 역은 홈 리스케줄링 프로세스가 없이 경로제어 프로세스가 스케줄에 대해서만 진로제어를 하고 있다. 이 운영중인 시스템에 다음과 같은 절차로 새로운 소프트웨어를 추가한다.

1) 프로세스 Customized

홈 리스케줄링 프로세스를 해당 역에 맞게 Customize한다. 홈 리스케줄링 프로세스에 필요한 역 정보들을 입력한다.

2) 프로세스 온라인 테스트

유지보수자가 유지보수용 컴퓨터에서 홈 리스케줄링 프로세스를 테스트모드로 가동한다. 이 테스트중인 프로세스는 온라인 데이터필드로 온라인 데이터를 수신하여 처리하고 홈 리스케줄링 결과를 테스트 데이터필드로 송출한다. 유지보수자는 테스트 데이터필드로 송출되는 결과들을 검토하거나 테스트 모드의 TMS를 가동하여 이 신규 프로세스

가 제대로 처리하고 있는가를 검증한다. 온라인/테스트 데이터필드 상의 데이터 확인을 위해서 유지보수자는 패킷 모니터링 툴을 사용할 수 있다.

3) 프로세스 배포

유지보수자는 홈 리스케줄링 프로세스와 이 프로세스에 대한 정보, 이 프로세스가 송수신하는 메시지를 등록한 프로파일들을 원격의 RCS서브시스템에 복사한다.

4) 프로세스 Reload

유지보수자는 웹을 통해서 RCS 서브시스템의 ADM (Autonomous Decentralized Manager)의 콘솔을 기동하여 프로세스 Reload를 명령한다. 이 과정에서 다른 서브시스템들 즉, 열차 추적(RTPS), 열차 모니터링(TMS) 서브시스템의 기능은 전혀 방해받지 않는다.

4.1.2 온라인상 애플리케이션 프로세스의 디버깅과 테스트 시나리오

Fig. 7과 같이 홈 리스케줄링 프로세스를 온라인 상에 추가하여 운영중인 상황이다. 이전에 단위 테스트와 온라인상 테스트를 수행하였지만 이때 홈 리스케줄링 프로세스의 동작에 문제를 발견되었고 유지보수자는 소프트웨어의 어느 부분을 변경하여야 할지 알 수 없었다고 가정하자. 자율역 제어 시스템에서 유지보수자는 유지보수용 컴퓨터에서 해당 프로세스를 현재 실시간 온라인 데이터를 수신하여 디버깅을 할 수 있다. 이때 다른 서브시스템들 열차 추적, 경로제어, 열차 모니터링의 기능은 전혀 방해받지 않는다.

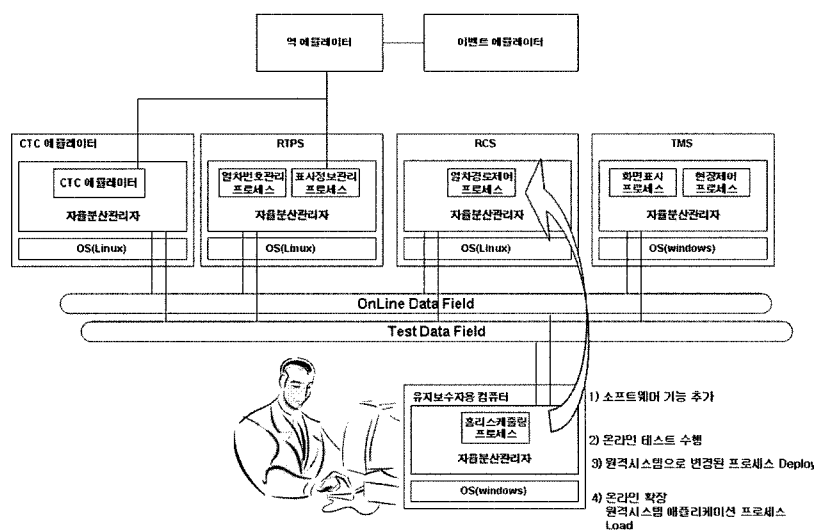


Fig. 6. Software Online Expansion

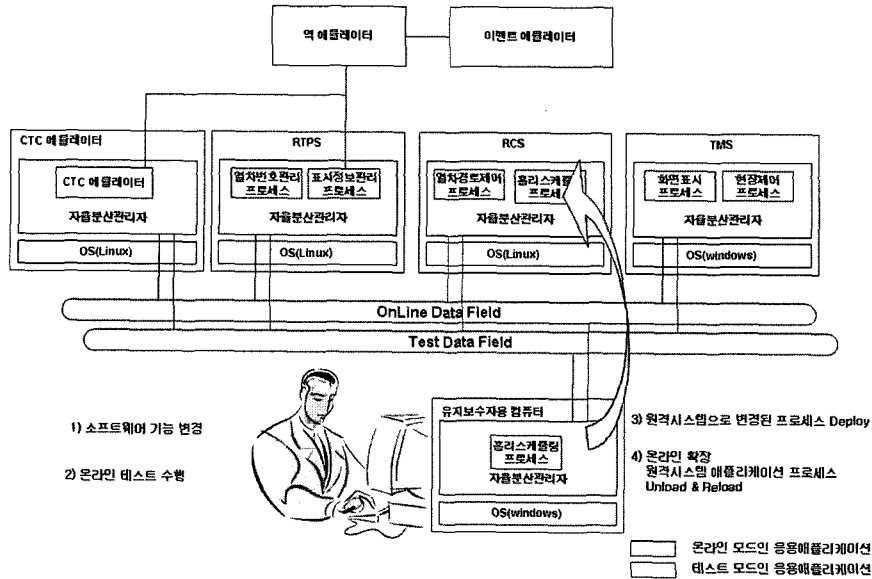


Fig. 7. Software Online Debugging and Online Testing

단지 RCS의 자율분산관리자를 기동하는 수 초간은 RCS가 담당한 리스케줄링과 경로제어는 수행되지 않는다. 중앙 집중 시스템의 관점에 보는 경우 이와 같은 상황은 라인전체의 열차추적, 경로제어, 열차 모니터링 등 전 기능과 전역에 대해서 영향을 미치는 게 되나, 자율분산시스템에서는 해당역, 해당 기능의 제약으로 영향 범위가 매우 작다.

4.2 단계적 구축(역 확장)

온라인 확장성은 역 확장과 소프트웨어 확장으로 나누어 볼 수 있다. 역 확장은 새로운 역에 자율분산 역 제어 시스템을 도입하는 경우이고, 소프트웨어 확장은 새로운 프로세스를 추가하는 경우를 들 수 있다. 여기서는 역 확장에 대해 모드의 변화와 동작 변화를 살펴보기로 한다.

자율분산 역 제어 시스템은 인접역 시스템의 유무, 센터의 유무에 따라서 역 단독 모드, 인접역 연동 모드, 센터 연동 모드로 달리 동작한다. 또한 이 환경 변화는 자동으로 감지되어 운영자에게 통보된다.

1) 역 단독 모드에서의 운영

라인 센터와 인접역 시스템이 구축되지 않은 상황에서 하나의 역에 자율 역 제어 시스템이 도입된 경우에 이 시스템은 역 단독 모드로 운영된다. 역 단독 모드에서는 외부로부터 스케줄을 수신하여 사용할 수 없으므로 해당 역의 시스템 내에서 스케줄을 작성하여 사용할 수 있어야 한다. TMS 서비스시스템은 역 단독 모드에서 해당역의 스케줄을 작성하도록 MMI(Man-Machine Interface)를 제공한다.

역 단독 모드에서는 인접역에서 발생하는 열차 정보가 없으므로 스케줄상의 정보에 근거하여 열차번호를 부여하게 된다.

2) 인접역 연동 모드에서의 운영

인접역에 새로 자율분산 역 제어 시스템이 도입되는 경우 역 단독 모드에서 운영되던 시스템은 그 환경변화를 감지하여 인접역 연동 모드로 모드 변경이 가능함을 운영자에게 통보한다. 운영자가 인접역 연동 모드를 선택하면 이때부터 시스템은 인접역 연동 모드로 동작한다.

인접역 연동 모드에서는 스케줄상의 열차번호와 인접역에서 송신한 열차번호간의 열차번호 대조를 한다. 두 열차번호가 일치하면 그대로 그 열차번호를 부여하고, 일치하지 않는 경우에는 역 운영자에게 질의하여 결정한 열차번호를 부여한다.

3) 센터 연동 모드에서의 운영

인접역 연동 모드로 운영하고 있는 역 시스템에 라인 센터가 구축되면 역 시스템은 이 환경 변화에 따라서 운영자에게 센터 연동 모드로 변경이 가능함을 통보한다. 운영자가 센터 연동 모드를 선택하면 이때부터 이 시스템은 센터 연동 모드로 동작한다. 센터 연동 모드에서는 본선 스케줄은 라인 센터에서 작성되어 송신된다. 자율 역 제어 시스템에서는 본선 열차의 도착/출발 스케줄, 구내 작업 스케줄에 대해서 작성한다.

4) 운영 모드의 감지

시스템은 인접역 시스템의 유무, 라인 센터의 유무와 같은 환경 변화에 따라서 운영 모드를 자동으로 감지한다. 그리고 선택 가능한 모드를 화면상에 표시한다. 운영자는 TMS 화면을 통해서, 또는 구축된 시스템의 정상 동작여부를 우선상으로 통보받으면, 정해진 업무 절차에 따라서 운영모드를 상향시킨다. 이것은 역 단독 모드에서 인접역 연동 모드로, 인접역 연동 모드에서 센터 연동 모드로의 전환을 말한다.

운영중에 시스템에 정보 송수신으로 협력하던 외부 시스템들(인접역 시스템, 센터)이 다운된 경우에는 모드 하향 전환을 하고 이를 경보로 표시하여 운영자에게 알린다. 센터 연동 모드에서 센터가 다운된 상황이 발생하면 인접역 연동 모드로 모드 하향 전환한다. 인접역 연동 모드에서 인접역이 다운된 경우에는 역 단독 모드로 하향 전환한다. 모드 하향 전환은 일종의 고장 대처 모드의 성격을 갖는다.

5. 결론

여기까지 자율 역 제어 시스템의 핵심 서브시스템의 설계와 단계적 구축과 온라인 속성에 대한 테스트 시나리오를 살펴보았다. 이 절에서는 이러한 단계적 구축과 온라인 속성 구현에 대해서 이 시스템이 자율분산시스템 기술을 어느 정도 구현하고 있는가, 시스템 장점과 앞으로 개선되어야 할 점을 살펴본다.

5.1 자율분산 시스템 기술의 적용 정도

앞서 자율분산 시스템의 서브시스템이 갖추어야 할 요건은 균질한 구조, 평등한 기능, 국소정보에 기초한다고 하였다. 이번에 설계한 자율 역 제어 시스템(Autonomous Station Control System : ASCS)는 역시 자율분산관리자라는 동일한 입출력구조를 가지므로 균질한 구조라고 할 수 있다. 각각의 서브시스템들은 다른 서브시스템으로 어떠한 명령을 내리거나 명령을 받는 구조가 아니라 데이터 드리븐 방식으로 동작하도록 설계되어 있다. 오직 수신한 메시지에 의해서만 정해진 동작을 수행하고 정해진 메시지를 주기적으로 송신하도록 되어 있다. 각각의 서브시스템의 목적을 달성하기 위해서 동작의 순서가 서브시스템의 내부에 결정되어 있지 않고 동작은 수신한 메시지의 종류와 내용에 달려있다. 다음으로 이 서브시스템은 국소정보에 기초한다. 인접역의 ASCS의 가동여부는 당역 ASCS의 가동에 큰 파급을 미치지 않는다. 인접역과 센터의 전체적인 정보의 존재는 당역 ASCS의 기능을 향상시키지만, 전체적인 정보의 부재시에

도 역별 시스템의 기능 수행에 장애를 가져오지는 않는다.

5.2 기존 시스템과의 차별성

여기서는 기존 중앙집중식의 열차 운행 제어 시스템에 비해서 이 자율분산 역 제어 시스템이 어떠한 장점을 가지는가에 대해서 살펴본다. 자율 역 제어 시스템의 기존 시스템과의 가장 큰 차별성은 응용 소프트웨어의 기능과 목적의 측면이 아니라. 비즈니스 로직을 담당하는 응용 소프트웨어와 통신과 애플리케이션 관리를 담당하는 자율분산관리자로 분리된 구조 자체, 시스템을 바라보는 관점이 제공하는 여러 가지 장점들에 있다고 할 수 있다. 이러한 장점으로써는 온라인 확장과 테스트, 단계적 구축을 들 수 있다. 역 시스템 단독으로도 운영하고, 인접역 시스템과도 연동하여 운영하고, 나중에 구축되는 라인센터와도 연동하여 운영하는 유연성 있는 아키텍처 자체에 장점이 있다. 자율적으로 자신의 환경에 맞춰 협조를 받는다. 협조는 받지만, 명령을 받지 않는다는 것이다. 협조가 없는 대로도 그 제약환경 안에서 운영한다. 그러한 제약에 의해서 일부 기능은 저하되지만 그 외의 다른 기능은 정상적인 운영이 가능한 것이다.

1) 자율분산시스템으로써의 시스템 장점

자율 역 제어 시스템의 비즈니스 로직의 차별성에서가 아니라 자율 분산 시스템의 구조 설계를 지향함으로써 얻어지는 장점들이 있다. 그것은 단계적 구축이 가능한 시스템이라는 점, 응용 소프트웨어의 온라인 추가와 온라인 테스트가 가능하다는 점이다. 단계적 구축의 방식은 일괄 구축 방식에 비해서 시스템 운영에 대한 리스크를 매우 감소시키고, 애플리케이션의 온라인 추가와 테스트는 중단 없는 미션 크리티컬 시스템의 운영을 가능하게 한다.

2) 사용자 관점에서의 기능 제공

이 시스템의 구조적인 장점 외에도 실제적으로 현업에 도움을 줄 수 있는 역 업무 자동화에 중점을 두어 운영팀장, 수송팀장, 신호팀장의 권한 위임과 작업 분배에 기여할 수 있다.

5.3 향후 발전 방향 및 개선점

현재 자율분산관리자는 통신 미들웨어의 기능과 애플리케이션 관리 기능을 모두 갖고 있다. 자율분산관리자는 프로세스 응용 애플리케이션의 리소스 관리(DB 커넥션, 프로세스 등), 웹 GUI상에서의 프로세스 정보 조회 및 Reload, 메시지 정보 조회 및 입력, 성능 모니터링 등 응용애플리케이션을 관리하는 기능을 확대하는 것이 시스템 운영, 보수

에 유용하다.

온라인 유지보수와 테스트, 온라인 확장을 위해서는 그 전제조건으로 응용애플리케이션이 서로 직접적인 연결을 갖지 않고 서로 의존적이지 않아야 한다. 현재의 프로세스보다는 더욱 적은 단위로 나누는 것이 완전한 위치 투명성, 데이터 드리븐 방식을 달성하기에 더 효율적일 것으로 보인다.

응용 소프트웨어가 관리된다는 것은 애플리케이션의 정보, 버전, 기능, 상태가 투명하게 표출될 수 있다는 것이다. 이는 비즈니스 기능의 확장성, 안정성, 이동성, 투명성을 확대할 수 있을 것이다. 철도역 열차 자동운행제어 시스템에 대한 자율분산 시스템 기술의 연구 진행이 다른 비즈니스 영역에 현존하는 최신의 연구, 기술들을 접목하는 작은 계기가 될 것으로 본다.

참 고 문 헌

1. 이영수, 안진, 김은희 (2005), “자율분산시스템 단계적 구축방식의 철도역 적용에 관한 연구”, 한국철도학회 춘계학술대회.
2. 홍순흠, 김영훈, 박범환 (2005), “대형고밀도 철도역 자동운행제어시스템을 위한 시뮬레이터 개발”, 대한전기학회 하계학술대회.
3. 홍순흠, 김영훈, 박범환 (2004), “고밀도 열차운행을 위한 트래픽 관리 방안 연구”, 대한전기학회 하계학술대회.
4. 김영훈, 홍순흠 (2005), “로컬트래픽을 고려한 역시스템 설계에 관한 연구” 한국철도학회 춘계학술대회.
5. 김영훈, 홍순흠 (2003), “열차운행관리시스템 비교 연구”, 한국철도학회, 추계철도학회.
6. 김영훈, 홍순흠, 정태운, 안진, 김윤희, 박성규 (2004), “자율분산형 열차운행제어체계에 관한 연구”, 한국철도학회, 춘계학술대회.
7. Mori, K. (1993), “Autonomous Decentralized Systems; Concept, Data Field Architecture and Future Trends”, Proc. Of ISADS93, pp.28-34.
8. Kitahara F. Kamijou, K. Kakurai, Y. Bekki, K. Kera, K. Kawano, K. (1999), “Phased-In Construction Method of ATOS”, IEEE, pp.415-424.
9. Kitahara, F. Kamijou, K. Kakurai, Y. Bekki, K. Kera, K. Kawano K. (1995), “Wide-Distributed Train-Traffic Computer Control system and Its Step-by-step Construction”, Proc. Of ISADS95, pp.311-318.
10. Kera K. Bekki, K. Fujiwara, K. Kamijou, K. Kitahara, F. (2001), Assurance Technology for Growing System and Its Application to Tokyo Metropolitan Railway Network, IEICE TRANS. INF. & SYST, pp.1085-1093.