

Unit Testing Plan

for Public Transport System

- Test Plan
- Test Design Specification
- Test Cases Specification

Project Team

Team 7

Date

2014-11-20

Team Information

박재원 201011332

김철진 201211032

장계인 201312412

Table of Contents

| | | |
|-----|--|----|
| 1 | Introduction | 4 |
| 1.1 | Objectives..... | 4 |
| 1.2 | Background | 4 |
| 1.3 | Scope..... | 4 |
| 1.4 | Project plan | 4 |
| 1.5 | Configuration management plan..... | 4 |
| 1.6 | References..... | 5 |
| 2 | Test items | 5 |
| 3 | Features to be tested..... | 12 |
| 4 | Features not to be tested | 15 |
| 5 | Approach..... | 17 |
| 6 | Item pass/fail criteria | 17 |
| 7 | Unit test design specification..... | 17 |
| 7.1 | Test design specification identifier | 17 |
| 7.2 | Features to be tested | 17 |
| 7.3 | Approach refinements..... | 17 |
| 7.4 | Test identification | 18 |
| 7.5 | Feature pass/fail criteria | 22 |
| 8 | Unit test case specification..... | 22 |
| 8.1 | Test case specification identifier | 22 |
| 8.2 | Test items | 26 |

8.3 Input specifications..... 26

8.4 Output specifications..... 27

9 Testing tasks 27

10 Environmental needs 27

11 Unit Test deliverables..... 27

12 Schedules 27

1 Introduction

1.1 Objectives

이 문서는 Team7의 Public Transport System의 Unit Test를 수행하기 위해 작성된 계획 문서이며, 본 System이 제대로 작동하는지를 살펴보기 위해 필요한 요소들을 정리해 놓은 문서이다. 앞선 단계에서 작성했던 Team7-2014.PTS.SRA-4.0의 State Transition Diagram을 바탕으로 Test할 부분을 선정하였고, 각 기능이 제대로 수행되는지 확인하는 것에 중점을 두어 작성하였다. Test를 수행하기 위해 필요한 활동 및 자원을 정의하고, Test Approach 및 Techniques를 정의한다. 또한 Test를 위한 환경적인 요구사항 및 Test 도구들을 정의한다.

1.2 Background

Public Transport System은 대중 교통 시스템으로 여기서는 소프트웨어만을 이용한 가상의 시스템으로 구현한다. 이 시스템은 지하철과 버스 단말기 및 정산 시스템으로 이루어져 있다. 텍스트 파일로 대체된 카드의 입력을 필요로 하며, 조건에 따라 결제가 이루어지고 하루 운행이 끝나면 지하철과 버스에 대한 정산이 이루어진다. Unit Test는 시스템을 구성하는 최소 단위 모듈들을 대상으로 하는 Test이며, 시스템의 성능을 좌우하는 모듈들이 요구사항을 만족하는지 확인 할 수 있는 Test Approach이다.

1.3 Scope

이 계획 문서는 Public Transport System (이하 PTS)의 Unit Test를 수행하기 위한 모든 것을 포함한다. PTS의 Unit Test를 수행하기 위한 자원과 절차, Test Approach와 Techniques과 필요로 하는 환경 및 도구 등을 정의한다. PTS의 Unit Test는 시스템을 구성하는 최소 단위의 모듈들을 대상으로 하며, 구현된 모듈이 요구사항을 만족하는지를 Test 한다.

1.4 Project plan

1.5 Configuration management plan

PTS의 Program Source Code 및 Unit Test를 위한 Test Code는 CTIP(Continuous Testing & Integration Platform) 환경에서 이루어지며, Program Source Code와 Test Code의 변경 및 수정사항은 지속적으로 통합되고 Test된다.

(1) Program Source Code의 변경

Program Source Code에 변경 및 수정사항 발생시, 이를 통합하고 수동적으로 Unit Test를 수행한다.

(2) 일정 주기

Program Source Code는 일정 주기를 가지고 Build 및 Unit Test를 수행한다.

1.6 References

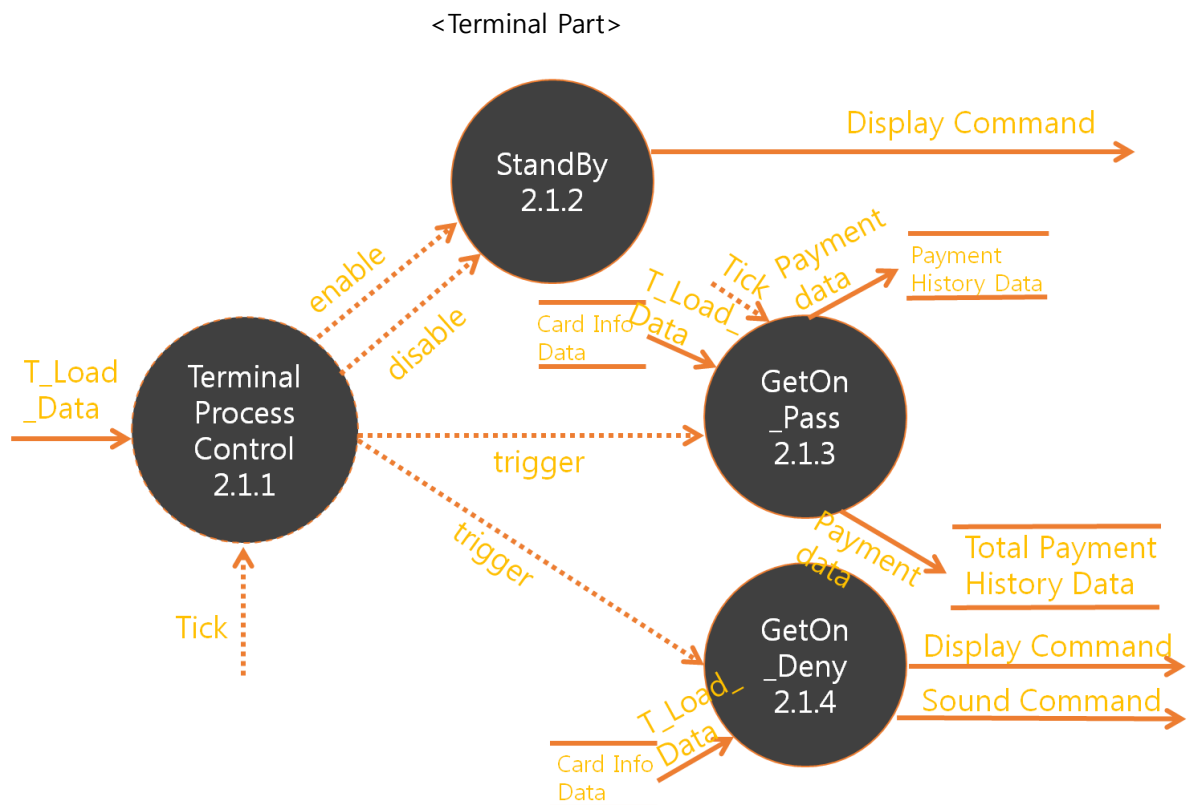
- Team7-2014.PTS.SRS-1.0
- Team7-2014.PTS.SRA-4.0
- Team7-2014.PTS.SDS-2.0

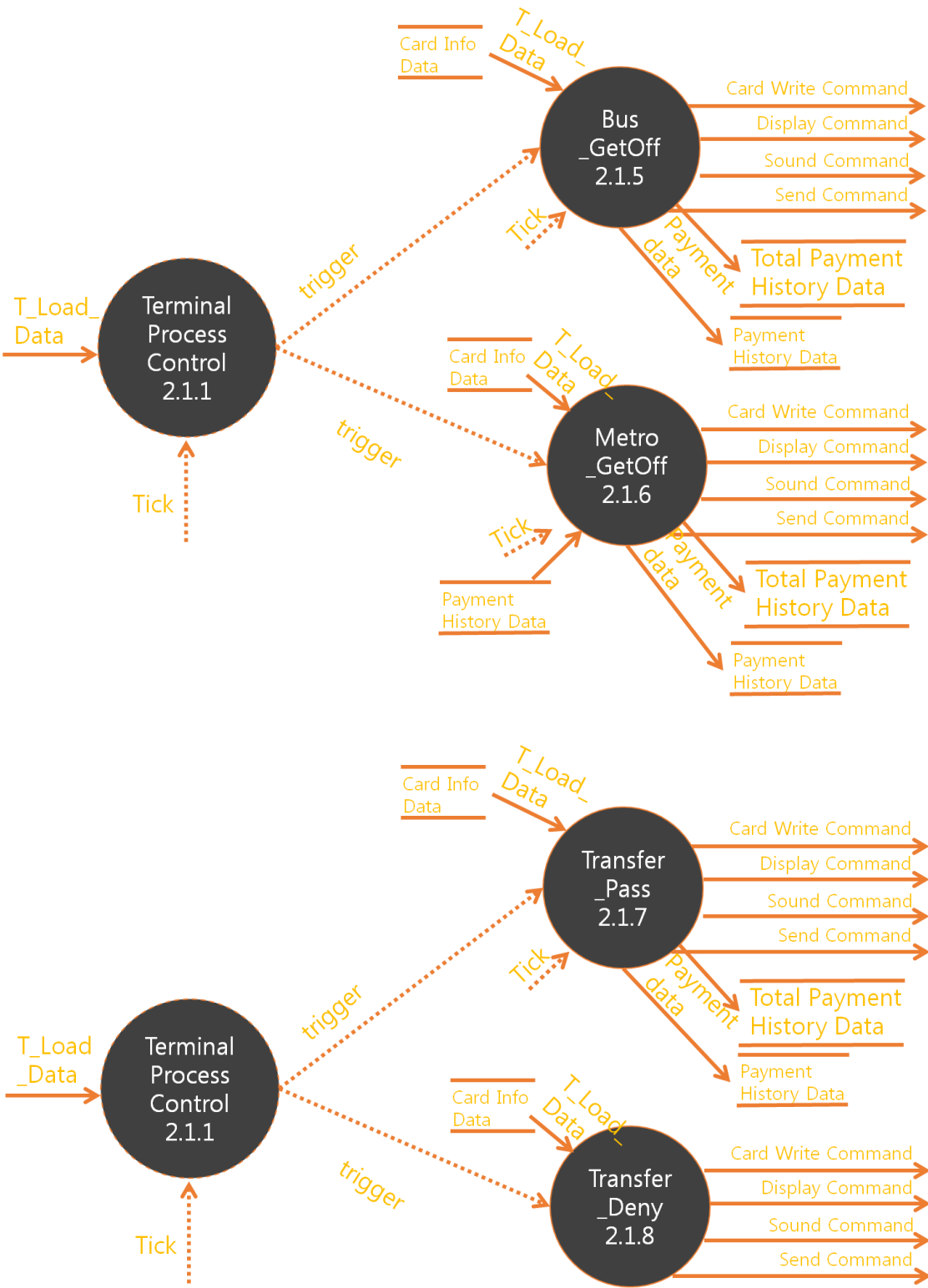
2 Test items

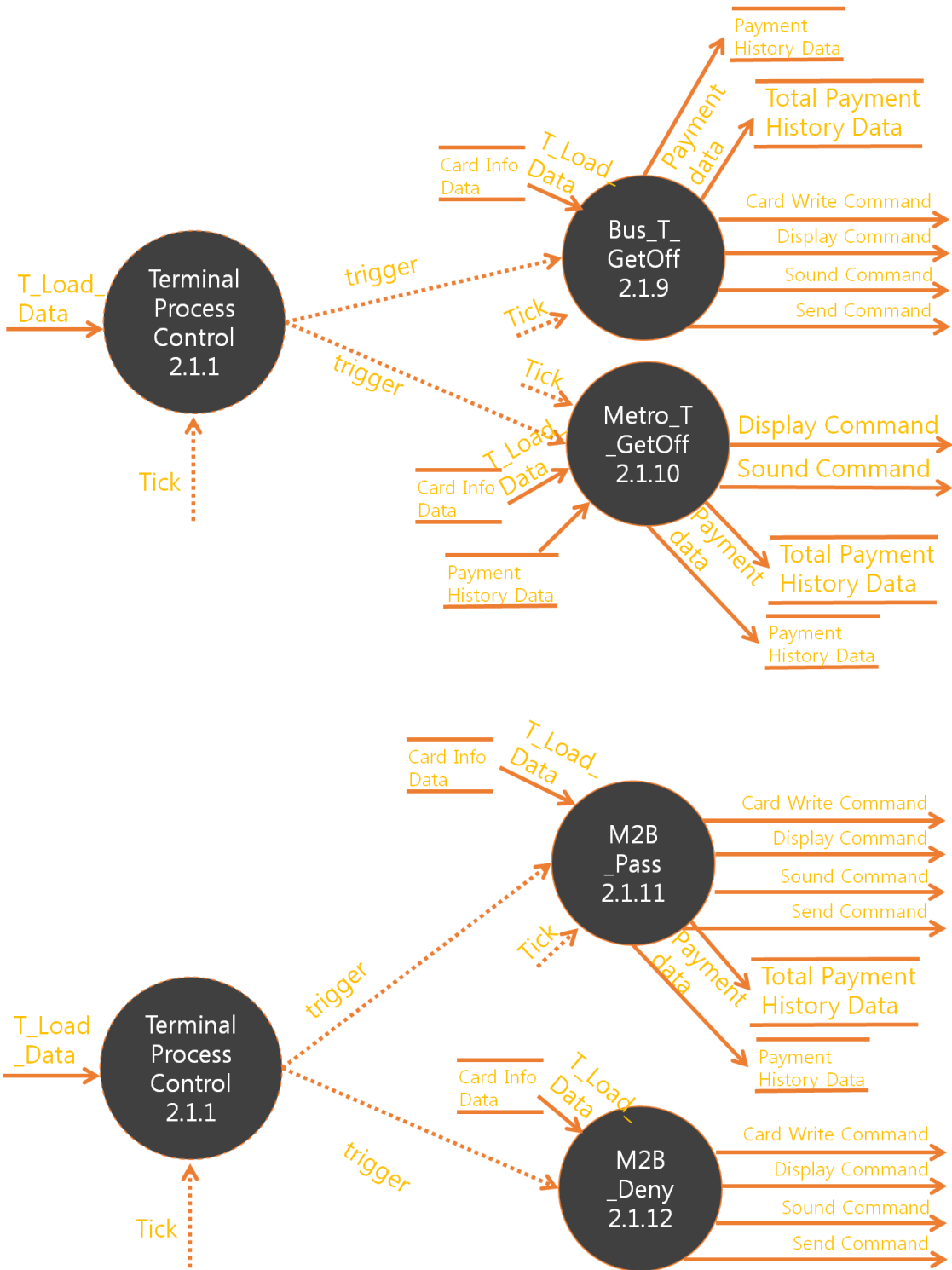
PTS를 구성하는 최소 단위 모듈들이 Unit Test의 대상이 된다. 각 모듈의 요구사항을 만족하는지 Test한다. Test Item은 다음 자료들로부터 작성되었다.

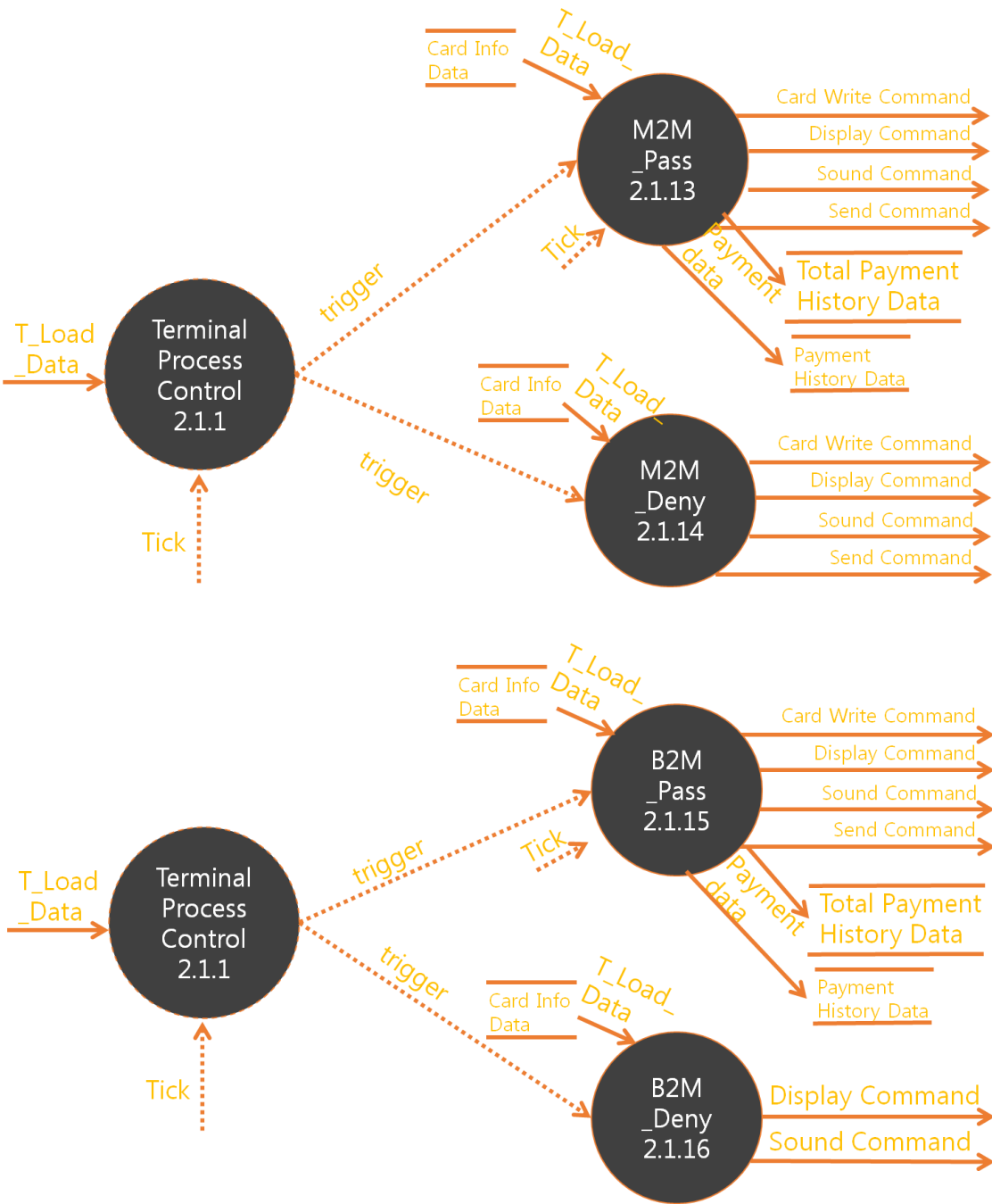
(1) Functionality of modules – Team7-2014.PTS.SRA-4.0: Process Specification

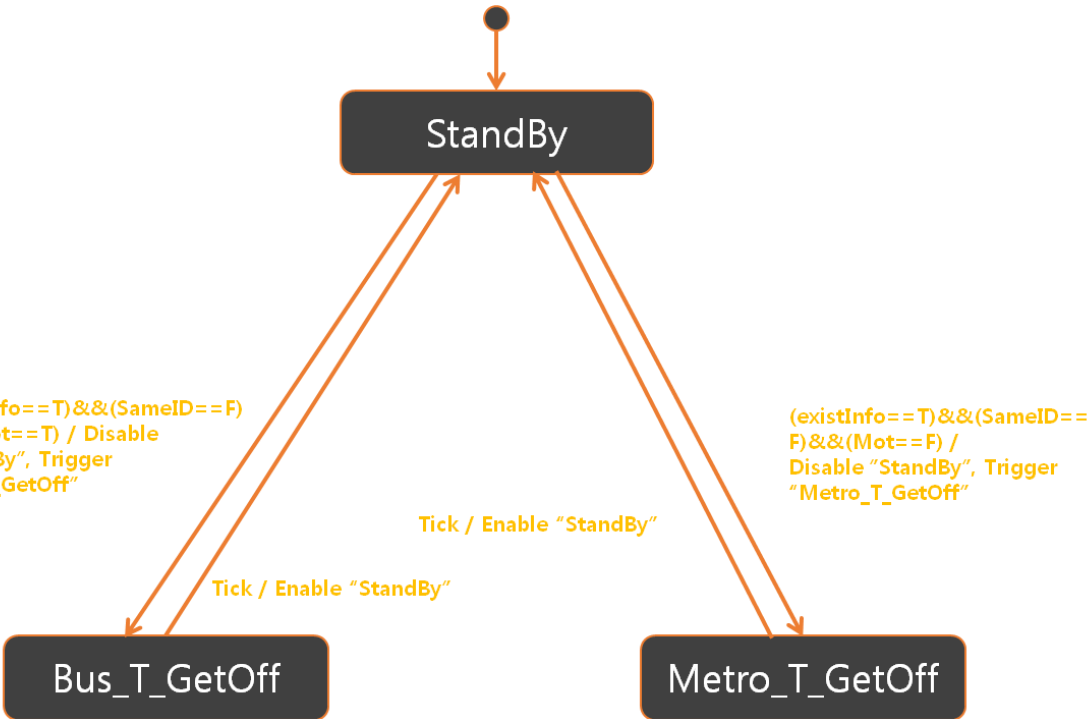
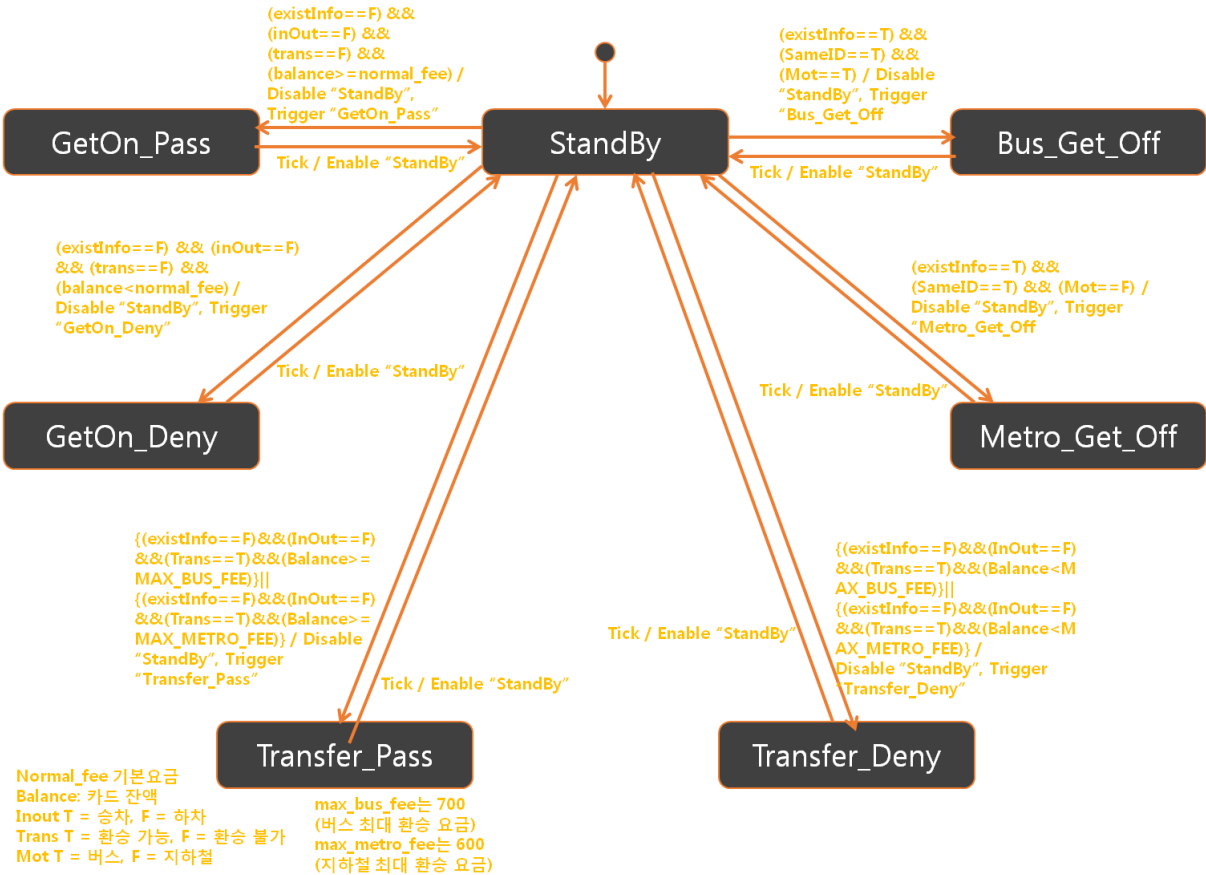
아래 그림은 일부를 참조한 것이다.

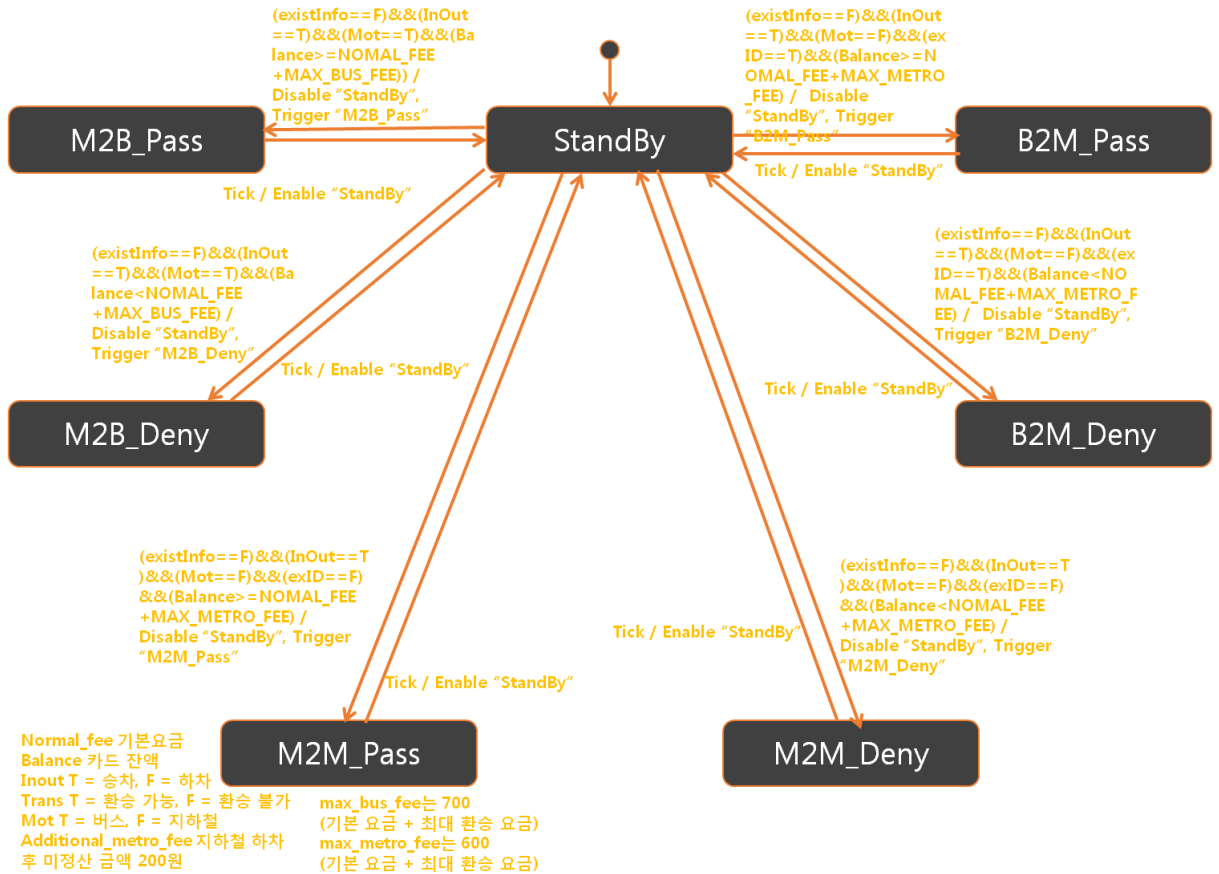






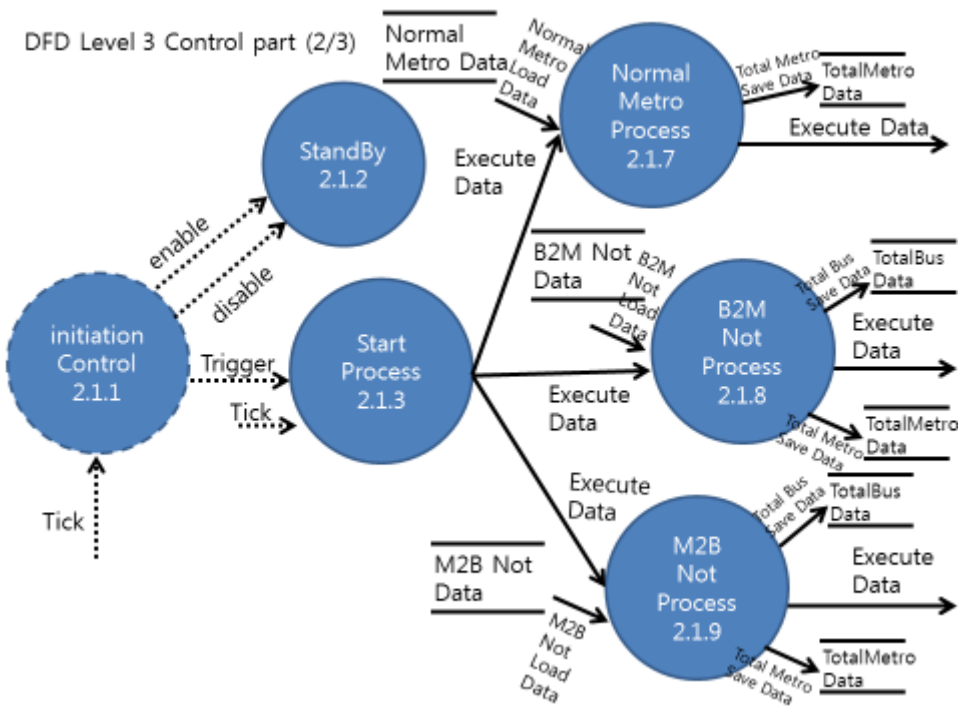
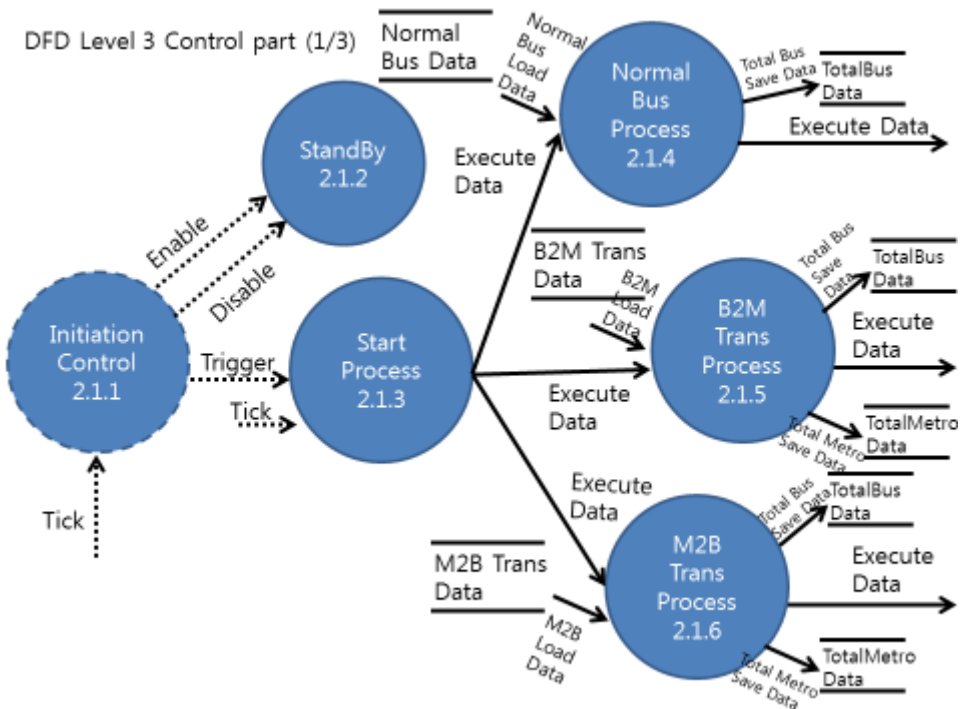






| | |
|---------------------|---|
| Reference No. | 2.1.1 |
| Name | Terminal Process Control |
| Input | T_Load Data, Tick |
| Output | Enable / Disable Command, Trigger Command |
| Process Description | Card-Info Store에서 카드 정보를 받아와, 각각의 시나리오를 만족하는 조건과 카드 정보를 비교하여 알맞은 프로세스에 트리거 명령을 보낸다. |

<Calculating System Part>



| | |
|---------------------|--|
| Reference No. | 2.1.1 |
| Name | Initiation Control |
| Input | Tick |
| Output | Enable, Disable, Trigger |
| Process Description | 컨트롤은 StandBy Enable 상태를 유지하고 있다가 하루(3분) 마다 Tick을 받는다. 이 때, StandBy 상태는 Disable 상태가 되고 Start Process를 Trigger 한다. |

(2) Module interface – Team7-.2014.PTS.SRA-4.0: Process specification, structure chart

3 Features to be tested

(1) Process in SRA: 각 프로세스가 가지고 있는 요구사항을 만족하는지 Test한다.

(2) Modules in SDS: 각 모듈이 가지고 있는 데이터 인터페이스를 Test한다. <Table 1 테스트할 Process(DFD) 리스트 (Terminal Part, Calculating System Part)>의 Process name 참조

<Table 1 테스트할 Process(DFD) 리스트 (Terminal Part)>

| ID | Name | Description |
|-------|--------------------------|--|
| 2.1.1 | Terminal Process Control | Card Info Data의 T_Load Data와 Digital Clock의 Tick(시간Data)를 받아온 후, 카드의 상태를 비교 판단하여 적절한 하위 프로세스로 Trigger명령을 보낸다.(Trigger시 StandBy의 상태를 Disable로 변경) |
| 2.1.3 | GetOn_Pass | 정상적인 탑승가능의 경우 실행되는 프로세스이다. Trigger 명령을 받으면, Card Info Data의 T_Load Data를 받아와 수정한 후 Card Write Command, Display Command, Sound Command, Send Command, Payment data 를 보낸다. |
| 2.1.4 | GetOn_Deny | 정상적인 탑승거부의 경우 실행되는 프로세스 Trigger 명령을 받으면, Card Info Data의 T_Load Data를 받아온 후 Display Command, Sound Command를 보낸다. |
| 2.1.5 | Bus_GetOff | 버스에서 하차하는 경우 실행되는 프로세스이다. Trigger 명령을 받으면, Card Info Data의 T_Load |

| | | |
|--------|----------------|---|
| | | Data를 받아와 수정한 후 Card Write Command, Display Command, Sound Command, Send Command, Payment data 를 보낸다. |
| 2.1.6 | Metro_GetOff | 지하철에서 하차하는 경우 실행되는 프로세스이다. Trigger 명령을 받으면, Total Payment History Data 와 Card Info Data의 T_Load Data를 받아와 수정한 후 Card Write Command, Display Command, Sound Command, Send Command를 보낸다. |
| 2.1.7 | Transfer_Pass | 환승 가능의 경우 실행되는 프로세스이다. Trigger 명령을 받으면, Card Info Data의 T_Load Data를 받아와 수정한 후 Card Write Command, Display Command, Sound Command, Send Command, Payment data 를 보낸다. |
| 2.1.8 | Transfer_Deny | 환승 거부의 경우 실행되는 프로세스이다. Trigger 명령을 받으면, Card Info Data의 T_Load Data를 받아온 후 Display Command, Sound Command를 보낸다. |
| 2.1.9 | Bus_T_GetOff | 버스로 환승 후 하차시 실행되는 프로세스이다. Trigger 명령을 받으면, Card Info Data의 T_Load Data를 받아와 수정한 후 Card Write Command, Display Command, Sound Command, Send Command, Payment data 를 보낸다. |
| 2.1.10 | Metro_T_GetOff | 지하철로 환승 후 하차시 실행되는 프로세스이다. Trigger 명령을 받으면, Total Payment History Data 와 Card Info Data의 T_Load Data를 받아와 수정한 후 Card Write Command, Display Command, Sound Command, Send Command, Payment data 를 보낸다. |
| 2.1.11 | M2B_Pass | 미정산(지하철에서 버스 환승 후 하차 시 단말기를 태그 하지 않았을 때) 탑승 가능의 경우 실행되는 프로세스이다. Trigger 명령을 받으면, Card Info Data의 T_Load Data를 받아와 수정한 후 Card Write Command, Display Command, Sound Command, Send Command, Payment data 를 보낸다. |
| 2.1.12 | M2B_Deny | 미정산(지하철에서 버스 환승 후 하차 시 단말기를 태그 하지 않았을 때) 탑승 거부의 경우 실행되는 프로세스이다. |

| | | |
|--------|----------|--|
| | | Trigger 명령을 받으면, Card Info Data의 T_Load Data를 받아온 후 Display Command, Sound Command를 보낸다 |
| 2.1.13 | M2M_Pass | 미정산(지하철에서 일반 하차 시 단말기를 태그 하지 않았을 때) 탑승 가능의 경우 실행되는 프로세스이다. Trigger 명령을 받으면, Card Info Data의 T_Load Data를 받아와 수정한 후 Card Write Command, Display Command, Sound Command, Send Command, Payment data 를 보낸다. |
| 2.1.14 | M2M_Deny | 미정산(지하철에서 일반 하차 시 단말기를 태그 하지 않았을 때) 탑승 거부의 경우 실행되는 프로세스이다. Trigger 명령을 받으면, Card Info Data의 T_Load Data를 받아온 후 Display Command, Sound Command를 보낸다 |
| 2.1.15 | B2M_Pass | 미정산(버스에서 지하철 환승 후 하차 시 단말기를 태그 하지 않았을 때) 탑승 가능의 경우 실행되는 프로세스이다. Trigger 명령을 받으면, Card Info Data의 T_Load Data를 받아와 수정한 후 Card Write Command, Display Command, Sound Command, Send Command, Payment data 를 보낸다. |
| 2.1.16 | B2M_Deny | 미정산(버스에서 지하철 환승 후 하차 시 단말기를 태그 하지 않았을 때) 탑승 거부의 경우 실행되는 프로세스이다. Trigger 명령을 받으면, Card Info Data의 T_Load Data를 받아온 후 Display Command, Sound Command를 보낸다. |

<Table 1 테스트할 Process(DFD) 리스트 (Calculating System Part)>

| ID | Name | Description |
|-------|--------------------|--|
| 2.1.1 | Initiation Control | StandBy의 상태를 Enable로 유지하고 있다가, 하루(3분)마다 Tick을 받는다. 이 때 StandBy상태를 Disable로 변경시키고 Start Process에 Trigger 명령을 보낸다. |
| 2.1.3 | Start Process | Trigger명령을 받으면 하위프로세스들 에게 Execute Data를 보낸다. |

4 Features not to be tested

- (1) Process in SRA: 외부 장치 드라이버, 단순 데이터 전달 프로세스 등은 제외한다.
- (2) Modules in SDS: <Table 2 테스트하지 않을 Process(DFD) 리스트 (Terminal Part, Calculating System Part)>의 Process name 참조

<Table 2 테스트하지 않을 Process(DFD) 리스트 (Terminal Part)>

| ID | Name | Description |
|-------|-----------------------|---|
| 1.1 | Card Reader Interface | Card로부터 들어온 Data를 Card Reader Manager에게 보낸다. |
| 1.2 | Card Reader Manager | Card Reader Interface로부터 받아온 Data를 Type별로 나눠서 Card Info Data로 보낸다. |
| 2.1.2 | StandBy | 카드가 태그 되지 않을 때의 상태로 입력이 들어올 때까지 정지 상태의 화면을 출력하며 상태를 유지한다. |
| 2.2.1 | Card Writer Interface | Card Info Data 의 T_Load Data를 받아와 Write Command의 Data와 비교 판단하여 History Output Data로 Save History Data를 보내고 Card로 Write Data를 보낸다. |
| 2.3.1 | Display Interface | Display Command를 받으면 단말기 화면에 띄워줄 Display Data를 보낸다. |
| 2.4.1 | Sound Interface | Sound Command를 받으면 단말기로 Sound Data를 보낸다. |
| 2.5.1 | Server Send Interface | Send Command를 받으면 History Output Data로부터 Load History Data를 받아와 서버(Calculating System)으로 Send Data를 보낸다. |

<Table 2 테스트하지 않을 Process(DFD) 리스트 (Calculating System Part)>

| ID | Name | Description |
|-------|-------------------------------|---|
| 1.1 | History Data Reader Interface | 단말기로부터 받은 Data를 History Data Reader Manager로 보낸다. |
| 1.2 | History Data Reader Manager | History Data Reader Interface로부터 받은 Data를 6가지 형태의 Data로 나눈 후 각각에 알맞은 저장소로 저장한다. |
| 2.1.2 | StandBy | 컨트롤러의 대기상태 프로세스이다 |
| 2.1.4 | Normal Bus Process | 버스의 일반 승차 경우 정산하는 프로세스이다. Normal Bus Load Data를 받아와 정산 처리 후 |

| | | |
|--------|-------------------------|--|
| | | Total Bus Data에 Total Bus Save Data를 보내고, End Process에 Execute EP Data를 보낸다. |
| 2.1.5 | B2M Trans Process | 버스에서 지하철로 환승 경우 정산하는 프로세스이다. B2M Trans Load Data를 받아와 정산 처리 후 Total Bus Data에 Total Bus Save Data를, Total Metro Data에 Total Metro Save Data를, 그리고 End Process에 Execute EP Data를 보낸다. |
| 2.1.6 | M2B Trans Process | 지하철에서 버스로 환승 경우 정산하는 프로세스이다. M2B Trans Load Data를 받아와 정산 처리 후 Total Bus Data에 Total Bus Save Data를, Total Metro Data에 Total Metro Save Data를, 그리고 End Process에 Execute EP Data를 보낸다. |
| 2.1.7 | Normal Metro Process | 지하철의 일반 승차 경우 정산하는 프로세스이다. Normal Metro Load Data를 받아와 정산 처리 후 Total Metro Data에 Total Metro Save Data를 보내고, End Process에 Execute EP Data를 보낸다. |
| 2.1.8 | B2M Not Process | 버스에서 지하철로 환승 한 후 하차태그를 찍지 않은 미정산 경우 정산하는 프로세스이다. M2B Trans Load Data를 받아와 정산 처리 후 Total Bus Data에 Total Bus Save Data를, Total Metro Data에 Total Metro Save Data를, 그리고 End Process에 Execute EP Data를 보낸다 |
| 2.1.9 | M2B Not Process | 지하철에서 버스로 환승 한 후 하차태그를 찍지 않은 미정산 경우 정산하는 프로세스이다. M2B Trans Load Data를 받아와 정산 처리 후 Total Bus Data에 Total Bus Save Data를, Total Metro Data에 Total Metro Save Data를, 그리고 End Process에 Execute EP Data를 보낸다 |
| 2.1.10 | End Process | Execute EP Data를 받으면 Complete Command, Display Command, Bus Payment Command, Metro Payment Command를 보낸다. |
| 2.2.1 | Bus Company Interface | Bus Payment Command를 받으면 Total Bus Load Data를 받아와 버스회사로 S_Send Data를 보낸다. |
| 2.3.1 | Metro Company Interface | Metro Payment Command를 받으면 Total Metro |

| | | |
|-------|--------------------|--|
| | | Load Data를 받아와 지하철회사로 S_Send Data를 보낸다. |
| 2.4.1 | Terminal Interface | Complete Command를 받으면 단말기에게 완료 신호(Signal Data)를 보낸다. |
| 2.5.1 | Display Interface | Display Command를 받으면 모니터로 S_Display Data를 보낸다. |

5 Approach

Public Transport System의 Program source code 및 Unit Test를 위한 Test code는 CTIP(Continuous Testing & Integration Platform)환경에서 이루어지며, Program source code 및 Test code의 변경 및 수정사항은 지속적으로 통합되고 Test된다.

(1) Brute force testing: 각 모듈의 요구사항을 만족하는지 확인할 수 있는 Test case를 작성한다. 그 이외의 예외사항에 대해서는 Test하지 않는다.

6 Item pass/fail criteria

Functional test pass/fail criteria: 각 모듈은 요구사항을 모두 만족하여야 한다.

7 Unit test design specification

7.1 Test design specification identifier

PTS.UTD.000.0000

7.2 Features to be tested

7.2.1 Process in SRA

<Table 1 테스트할 Process(DFD) 리스트 (Terminal Part, Calculating System Part)>
참조

7.3 Approach refinements

7.3.1 Brute force testing

PTS의 각 모듈이 요구사항을 만족하는지를 확인하기 위하여, 요구사항에 정의된 내용에 기반하여 Test Case를 작성한다. 그 이외의 예외 상황에 대해서는 작성하지 않는다.

7.4 Test identification

7.4.1 <Table 3 Test Design Identification (Terminal Part)>

| Identifier | Feature | Valid/Invalid value |
|------------------|--------------------------------|---|
| PTS.UTC_211_000 | 2.1.1 Terminal Process Control | StandBy==Enable 일 때, existInfo==F, inOut==F, trans==F, balance>=normal_fee인 경우 |
| PTS.UTC_211_001 | 2.1.1 Terminal Process Control | StandBy==Enable 일 때, existInfo==F, inOut==F, trans==F, balance<normal_fee 인 경우 |
| PTS.UTC_211_002 | 2.1.1 Terminal Process Control | StandBy==Enable 일 때, existInfo==F, InOut==F, Trans==T, Balance>=MAX_BUS_FEE인 경우 |
| PTS.UTC_211_003 | 2.1.1 Terminal Process Control | StandBy==Enable 일 때, existInfo==F, InOut==F, Trans==T, Balance>=MAX_METRO_FEE인 경우 |
| PTS.UTC_211_004 | 2.1.1 Terminal Process Control | StandBy==Enable 일 때, existInfo==F, InOut==T, Trans==T, Balance>=MAX_BUS_FEE인 경우 |
| PTS.UTC_211_005 | 2.1.1 Terminal Process Control | StandBy==Enable 일 때, existInfo==F, InOut==T, Trans==T, Balance>=MAX_METRO_FEE인 경우 |
| PTS.UTC_211_006 | 2.1.1 Terminal Process Control | StandBy==Enable 일 때, existInfo==F, InOut==F, Trans==T, Balance<MAX_BUS_FEE인 경우 |
| PTS.UTC_211_007 | 2.1.1 Terminal Process Control | StandBy==Enable 일 때, existInfo==F, InOut==F, Trans==T, Balance<MAX_METRO_FEE인 경우 |
| PTS.UTC_211_008 | 2.1.1 Terminal Process Control | StandBy==Enable 일 때, existInfo==F, InOut==T, Trans==T, Balance<MAX_BUS_FEE인 경우 |
| PTS.UTC_211_009 | 2.1.1 Terminal Process Control | StandBy==Enable 일 때, existInfo==F, InOut==T, Trans==T, Balance<MAX_METRO_FEE인 경우 |
| PTS.UTC_211_0010 | 2.1.1 Terminal Process Control | StandBy==Enable 일 때, existInfo==T, SameID==T, Mot==T인 경우 |
| PTS.UTC_211_0011 | 2.1.1 Terminal Process Control | StandBy==Enable 일 때, existInfo==T, SameID==T, Mot==F인 경우 |

| | | |
|------------------|--------------------------------|--|
| PTS.UTC_211_0012 | 2.1.1 Terminal Process Control | StandBy==Enable 일 때, existInfo==T, SameID==F, Mot==T인 경우 |
| PTS.UTC_211_0013 | 2.1.1 Terminal Process Control | StandBy==Enable 일 때, existInfo==T, SameID==F, Mot==F인 경우 |
| PTS.UTC_211_0014 | 2.1.1 Terminal Process Control | StandBy==Enable 일 때, existInfo==F, InOut==T, Mot==T, Balance>=NOMAL_FEE + MAX_BUS_FEE인 경우 |
| PTS.UTC_211_0015 | 2.1.1 Terminal Process Control | StandBy==Enable 일 때, existInfo==T, InOut==T, Mot==T, Balance>=NOMAL_FEE + MAX_BUS_FEE인 경우 |
| PTS.UTC_211_0016 | 2.1.1 Terminal Process Control | StandBy==Enable 일 때, existInfo==F, InOut==T, Mot==T, Balance<NOMAL_FEE + MAX_BUS_FEE인 경우 |
| PTS.UTC_211_0017 | 2.1.1 Terminal Process Control | StandBy==Enable 일 때, existInfo==T, InOut==T, Mot==T, Balance<NOMAL_FEE + MAX_BUS_FEE인 경우 |
| PTS.UTC_211_0018 | 2.1.1 Terminal Process Control | StandBy==Enable 일 때, existInfo==F, InOut==T, Mot==F, exID==F, Balance>=NOMAL_FEE + MAX_METRO_FEE인 경우 |
| PTS.UTC_211_0019 | 2.1.1 Terminal Process Control | StandBy==Enable 일 때, existInfo==T, InOut==T, Mot==F, exID==F, Balance>=NOMAL_FEE + MAX_METRO_FEE인 경우 |
| PTS.UTC_211_0020 | 2.1.1 Terminal Process Control | StandBy==Enable 일 때, existInfo==F, InOut==T, Mot==F, exID==F, Balance<NOMAL_FEE + MAX_METRO_FEE인 경우 |
| PTS.UTC_211_0021 | 2.1.1 Terminal Process Control | StandBy==Enable 일 때, existInfo==T, InOut==T, Mot==F, exID==F, Balance<NOMAL_FEE + MAX_METRO_FEE인 경우 |
| PTS.UTC_211_0022 | 2.1.1 Terminal Process Control | StandBy==Enable 일 때, existInfo==F, InOut==T, Mot==F, |

| | | |
|------------------|--------------------------------|---|
| | | exID==T, Balance>=NOMAL_FEE + MAX_METRO_FEE인 경우 |
| PTS.UTC_211_0023 | 2.1.1 Terminal Process Control | StandBy==Enable 일 때, existInfo==T, InOut==T, Mot==F, exID==T, Balance>=NOMAL_FEE + MAX_METRO_FEE인 경우 |
| PTS.UTC_211_0024 | 2.1.1 Terminal Process Control | StandBy==Enable 일 때, existInfo==F, InOut==T, Mot==F, exID==T, Balance<NOMAL_FEE + MAX_METRO_FEE인 경우 |
| PTS.UTC_211_0025 | 2.1.1 Terminal Process Control | StandBy==Enable 일 때, existInfo==T, InOut==T, Mot==F, exID==T, Balance<NOMAL_FEE + MAX_METRO_FEE인 경우 |
| PTS.UTC_213_0000 | 2.1.3 GetOn_Pass | Trigger 입력이 들어오면, T_Load Data를 받아와 Data 수정 후 결제 내역에 대한 Payment Data와 적절한 Command Data를 출력한다. |
| PTS.UTC_214_0000 | 2.1.4 GetOn_Deny | Trigger 입력이 들어오면, T_Load Data를 받아와 Data 수정 후 적절한 Command Data를 출력한다. |
| PTS.UTC_215_0000 | 2.1.5 Bus_GetOff | Trigger 입력이 들어오면, T_Load Data를 받아와 Data 수정 후 결제 내역에 대한 Payment Data와 적절한 Command Data를 출력한다. |
| PTS.UTC_216_0000 | 2.1.6 Metro_GetOff | Trigger 입력이 들어오면, Payment History Data와 T_Load Data를 받아와 Data 수정 후 결제 내역에 대한 Payment Data와 적절한 Command Data를 출력한다. |
| PTS.UTC_217_0000 | 2.1.7 Transfer_Pass | Trigger 입력이 들어오면, T_Load Data를 받아와 Data 수정 후 결제 내역에 대한 Payment Data와 적절한 Command Data를 출력한다. |
| PTS.UTC_218_0000 | 2.1.8 Transfer_Deny | Trigger 입력이 들어오면, T_Load Data를 받아와 Data 수정 후 적절한 Command Data를 출력한다. |
| PTS.UTC_219_0000 | 2.1.9 Bus_T_GetOff | Trigger 입력이 들어오면, T_Load Data를 |

| | | |
|-------------------|-----------------------|--|
| | | 받아와 Data 수정 후 결제 내역에 대한 Payment Data와 적절한 Command Data를 출력한다. |
| PTS.UTC_2110_0000 | 2.1.10 Metro_T_GetOff | Trigger 입력이 들어오면, Payment History Data와 T_Load Data를 받아와 Data 수정 후 결제 내역에 대한 Payment Data와 적절한 Command Data를 출력한다. |
| PTS.UTC_2111_0000 | 2.1.11 M2B_Pass | Trigger 입력이 들어오면, T_Load Data를 받아와 Data 수정 후 결제 내역에 대한 Payment Data와 적절한 Command Data를 출력한다. |
| PTS.UTC_2112_0000 | 2.1.12 M2B_Deny | Trigger 입력이 들어오면, T_Load Data를 받아와 Data 수정 후 적절한 Command Data를 출력한다. |
| PTS.UTC_2113_0000 | 2.1.13 M2M_Pass | Trigger 입력이 들어오면, T_Load Data를 받아와 Data 수정 후 결제 내역에 대한 Payment Data와 적절한 Command Data를 출력한다. |
| PTS.UTC_2114_0000 | 2.1.14 M2M_Deny | Trigger 입력이 들어오면, T_Load Data를 받아와 Data 수정 후 적절한 Command Data를 출력한다. |
| PTS.UTC_2115_0000 | 2.1.15 B2M_Pass | Trigger 입력이 들어오면, T_Load Data를 받아와 Data 수정 후 결제 내역에 대한 Payment Data와 적절한 Command Data를 출력한다. |
| PTS.UTC_2116_0000 | 2.1.16 B2M_Deny | Trigger 입력이 들어오면, T_Load Data를 받아와 Data 수정 후 적절한 Command Data를 출력한다. |

7.4.2 <Table 3 Test Design Identification (Calculating System Part)>

| Identifier | Feature | Valid/Invalid value |
|-----------------|--------------------------|--|
| PTS.UTC_211_000 | 2.1.1 Initiation Control | StandBy상태에서 Tick을 받는다. |
| PTS.UTC_211_001 | 2.1.1 Initiation Control | Start Process상태에서 Tick을 받는다. |
| PTS.UTC_213_000 | 2.1.3 Start Process | Trigger입력이 들어오면, 6개의 각 프로세스로 Execute Data를 전송한다. |

7.5 Feature pass/fail criteria

PTS의 각 모듈(프로세스)은 SRA에 정의되어 있는 요구사항 (입/출력 및 동작)을 모두 만족해야 한다. 각 모듈(프로세스)의 입/출력 및 동작은 SRA의 Process Description항목 및 State Transition Diagram을 참조한다.

8 Unit test case specification

8.1 Test case specification identifier

<Table 4 Test Case Identification (Terminal Part)>

| Test Case Identifier | Input Specification | Output Specification |
|----------------------|--|--|
| PTS.UTC_211_000 | StandBy==Enable / existInfo==F, inOut==F, trans==F, balance>=normal_fee | StandBy==Disable / Trigger -> GetOn_Pass |
| PTS.UTC_211_001 | StandBy==Enable / existInfo==F, inOut==F, trans==F, balance<normal_fee | StandBy==Disable / Trigger -> GetOn_Deny |
| PTS.UTC_211_002 | StandBy==Enable / existInfo==F, InOut==F, Trans==T, Balance>=MAX_BUS_FEE | StandBy==Disable / Trigger -> Transfer_Pass |
| PTS.UTC_211_003 | StandBy==Enable / existInfo==F, InOut==F, Trans==T, Balance>=MAX_METRO_FEE | StandBy==Disable / Trigger -> Transfer_Pass |
| PTS.UTC_211_004 | StandBy==Enable / existInfo==F, InOut==T, Trans==T, Balance>=MAX_BUS_FEE | StandBy==Enable |
| PTS.UTC_211_005 | StandBy==Enable / existInfo==F, InOut==T, Trans==T, Balance>=MAX_METRO_FEE | StandBy==Enable |
| PTS.UTC_211_006 | StandBy==Enable / existInfo==F, InOut==F, Trans==T, Balance<MAX_BUS_FEE | StandBy==Disable / Trigger -> Transfer_Deny |
| PTS.UTC_211_007 | StandBy==Enable / existInfo==F, InOut==F, Trans==T, Balance<MAX_METRO_FEE | StandBy==Disable / Trigger -> Transfer_Deny |
| PTS.UTC_211_008 | StandBy==Enable / existInfo==F, InOut==T, Trans==T, | StandBy==Enable |

| | | |
|------------------|--|---|
| | Balance<MAX_BUS_FEE | |
| PTS.UTC_211_009 | StandBy==Enable / existInfo==F, InOut==T, Trans==T, Balance<MAX_METRO_FEE | StandBy==Enable |
| PTS.UTC_211_0010 | StandBy==Enable / existInfo==T, SameID==T, Mot==T | StandBy==Disable / Trigger -> Bus_GetOff |
| PTS.UTC_211_0011 | StandBy==Enable / existInfo==T, SameID==T, Mot==F | StandBy==Disable / Trigger -> Metro_GetOff |
| PTS.UTC_211_0012 | StandBy==Enable / existInfo==T, SameID==F, Mot==T | StandBy==Disable / Trigger -> Bus_T_GetOff |
| PTS.UTC_211_0013 | StandBy==Enable / existInfo==T, SameID==F, Mot==F | StandBy==Disable / Trigger -> Metro_T_GetOff |
| PTS.UTC_211_0014 | StandBy==Enable / existInfo==F, InOut==T, Mot==T, Balance>=NOMAL_FEE + MAX_BUS_FEE | StandBy==Disable / Trigger -> M2B_Pass |
| PTS.UTC_211_0015 | StandBy==Enable / existInfo==T, InOut==T, Mot==T, Balance>=NOMAL_FEE + MAX_BUS_FEE | StandBy==Enable |
| PTS.UTC_211_0016 | StandBy==Enable / existInfo==F, InOut==T, Mot==T, Balance<NOMAL_FEE + MAX_BUS_FEE | StandBy==Disable / Trigger -> M2B_Deny |
| PTS.UTC_211_0017 | StandBy==Enable / existInfo==T, InOut==T, Mot==T, Balance<NOMAL_FEE + MAX_BUS_FEE | StandBy==Enable |
| PTS.UTC_211_0018 | StandBy==Enable / existInfo==F, InOut==T, Mot==F, exID==F, Balance>=NOMAL_FEE + MAX_METRO_FEE | StandBy==Disable / Trigger -> M2M_Pass |
| PTS.UTC_211_0019 | StandBy==Enable / existInfo==T, InOut==T, Mot==F, exID==F, Balance>=NOMAL_FEE + MAX_METRO_FEE | StandBy==Enable |
| PTS.UTC_211_0020 | StandBy==Enable / existInfo==F, InOut==T, Mot==F, exID==F, Balance<NOMAL_FEE + MAX_METRO_FEE | StandBy==Disable / Trigger -> M2M_Deny |
| PTS.UTC_211_0021 | StandBy==Enable / existInfo==T, InOut==T, Mot==F, | StandBy==Enable |

| | | |
|------------------|---|--|
| | exID==F, Balance<NOMAL_FEE + MAX_METRO_FEE | |
| PTS.UTC_211_0022 | StandBy==Enable / existInfo==F, InOut==T, Mot==F, exID==T, Balance>=NOMAL_FEE + MAX_METRO_FEE | StandBy==Disable / Trigger -> B2M_Pass |
| PTS.UTC_211_0023 | StandBy==Enable / existInfo==T, InOut==T, Mot==F, exID==T, Balance>=NOMAL_FEE + MAX_METRO_FEE | StandBy==Enable |
| PTS.UTC_211_0024 | StandBy==Enable / existInfo==F, InOut==T, Mot==F, exID==T, Balance<NOMAL_FEE + MAX_METRO_FEE | StandBy==Disable / Trigger -> B2M_Deny |
| PTS.UTC_211_0025 | StandBy==Enable / existInfo==T, InOut==T, Mot==F, exID==T, Balance<NOMAL_FEE + MAX_METRO_FEE | StandBy==Enable |
| PTS.UTC_213_0000 | Trigger in / T_Load_Data | Balance-=NORMAL_FEE, Inout==True, Mot==True(or False)(Current時刻) / Card Write Command, Display Command, Sound Command, Send Command Payment data |
| PTS.UTC_214_0000 | Trigger in / T_Load_Data | Display Command, Sound Command |
| PTS.UTC_215_0000 | Trigger in / T_Load_Data. | Balance InOut==False Card Write Command, Display Command, Sound Command, Send Command, Payment data |
| PTS.UTC_216_0000 | Trigger in / T_Load_Data / Payment History Data | Balance InOut==False Card Write Command, |

| | | |
|-------------------|--|--|
| | | Display Command, Sound Command, Send Command, Payment data |
| PTS.UTC_217_0000 | Trigger in / T_Load_Data | Balance-=MAX_BUS_FEE or MAX_METRO_FEE, Inout==True, Mot==True(or False)(Current _額) / Card Write Command, Display Command, Sound Command, Send Command Payment data |
| PTS.UTC_218_0000 | Trigger in / T_Load_Data | Display Command, Sound Command |
| PTS.UTC_219_0000 | Trigger in / T_Load_Data | Balance InOut==False Card Write Command, Display Command, Sound Command, Send Command, Payment data |
| PTS.UTC_2110_0000 | Trigger in / T_Load_Data / Payment History Data | Balance InOut==False Card Write Command, Display Command, Sound Command, Send Command, Payment data |
| PTS.UTC_2111_0000 | Trigger in / T_Load_Data | Balance-=(NOMAL_FEE +MAX_BUS_FEE), Inout==True, Mot==True(or False)(Current _額) / Card Write Command, Display Command, Sound Command, Send Command Payment data |

| | | |
|-------------------|--------------------------|--|
| PTS.UTC_2112_0000 | Trigger in / T_Load_Data | Display Command, Sound Command, |
| PTS.UTC_2113_0000 | Trigger in / T_Load_Data | Balance-=(NOMAL_FEE +ADDITIONAL_METRO_FEE), InOut==True, Mot==True(or False)(Current값) / Card Write Command, Display Command, Sound Command, Send Command Payment data |
| PTS.UTC_2114_0000 | Trigger in / T_Load_Data | Display Command, Sound Command, |
| PTS.UTC_2115_0000 | Trigger in / T_Load_Data | Balance-=(NOMAL_FEE +MAX_METRO_FEE), Inout==True, Mot==True(or False)(Current값) / Card Write Command, Display Command, Sound Command, Send Command Payment data |
| PTS.UTC_2116_0000 | Trigger in / T_Load_Data | Display Command, Sound Command, |

<Table 4 Test case Identification (Calculating System Part)>

| Test Case Identifier | Input Specification | Output Specification |
|----------------------|---------------------|--|
| PTS.UTC_211_000 | Tick | StandBy==Disable / Trigger -> Start Process |
| PTS.UTC_211_001 | Tick | StandBy==Enable |
| PTS.UTC_213_002 | Trigger | Execute Data |

8.2 Test items

<Table 3 Test Design Identification (Terminal Part, Calculating System Part)> 참조

8.3 Input specifications

<Table 4 Test case Identification (Terminal Part, Calculating System Part)> 참조

8.4 Output specifications

<Table 4 Test case Identification (Terminal Part, Calculating System Part)> 참조

9 Testing tasks

<Table 5 Testing Tasks & Schedule>

| Task | Predecessor Tasks | Special Skills | Effort | Findish Date |
|-------------------------------|--------------------------------------|----------------|--------|--------------|
| (1) Unit Test Plan 작성 | SRS 작성 SRA 작성 SDS 작성 PTS 구현 | | 5 | |
| (2) Test Design Specification | Task 1 | PTS에 대한 이해 | 4 | |
| (3) Test Case specification | Task 2 | PTS에 대한 이해 | 4 | |
| (4) Test Execution | Task 3 | Test Code 작성 | 3 | |
| (5) Test Result Report | Task 4 | | 1 | |
| (6) 개발팀에게 Report 전달 | Task 5 | | 1 | |

10 Environmental needs

PTS의 Unit Test를 위한 환경적 요구사항은 다음과 같다.

- (1) Hardware & Platform, Visual Studio IDE (Integrated Development Environment)
MSVC Compiler / Linker
- (2) CTIP (Continuous Testing & Integration Platform) Environment

11 Unit Test deliverables

- 11.1 Unit test plan
- 11.2 Unit test design specification
- 11.3 Unit test case specification
- 11.4 Unit test summary report

12 Schedules

<Table 5 Testing Tasks & Schedule> 참조