Software Requirement Analysis
for Public Transportation System

Project Team

Team 4

Date

2014-09-25

노은방 200811428
김상민 200910044
박수민 201111353
한별 201214217

Table of Contents

목차

1 Introduction 3

1.1 Purpose 3

1.2 Scope 3

1.2.1 개발팀 3

1.2.2 제한사항 3

1.2.3 제품의 활용도 3

1.2.4 개발환경 4

1.3 Definition`, acronyms, and abbrevations 4

1.4 Reference 5

2 Overall Description 5

2.1 Product Perspective 5

2.2 Product functions 5

2.2.1 버스, 지하철 단말기 5

2.2.2 정산 5

2.3 User characteristics 5

2.4 Constraints 5

2.5 Assumptions and dependencies 5

3 Structured Analysis 6

3.1 System Context Diagram 6

3.1.1 Basic System Context Diagram....6
3.1.2 The System Context Diagram....7
3.2 Data Flow Diagram....7
3.2.1 DFD level 0....7
3.2.2 DFD Level 1....10
3.2.3 DFD Level 2....11
3.2.4 DFD Level 3....14
3.2.5 DFD Level 4....25
3.2.6 Overall DFD....28

1 Introduction

1.1 Purpose

Public Transportation System 에 관한 요구사항의 명세를 기반으로, Structured Analysis 를 수행한 문서이다. 이 문서를 통해 PTS 의 필요한 요구사항들의 관계를 명확히 하고, 더 나아가 이것은 실제 구현의 토대가 된다.

이 문서를 읽는 사람에게 PTS 의 Analysis 가 어떻게 이루어졌는지 명확히 전달한다.

1.2 Scope

1.2.1 개발팀

Team 4

1.2.2 제한사항

HW(단말기)와 연동까지 고려하지 않고, SW 로만 구동할 수 있도록 한다.

1.2.3 제품의 활용도

개발이 완료된 후 실제 교통 단말기와 교통 카드의 SW 를 개발하기 위한 프로토타입으로 삼을 수 있다.

1.2.4 개발환경

Cygwin,vi,gcc

1.3 Definition`, acronyms, and abbrevations

SW: Software HW: Hardware PTS: Public Transportation System

요금분류 :환승과 정산여부에 따라 요금계산을 나누어준다

저장소 비우기: 단말기에서 저장된 정보들을 초기화하는 프로세스이다

기본요금 : 각 버스와 지하철에서 추가요금 없이 탑승시 내야되는 요금이다

미정산 환승 : 이전 탑승수단에서 태그를 안하고 탑승시 수행하는 금액계산이다

정산환승 : 이전 탑승수단에서 태그를 한 경우 탑승시 수행하는 금액계산이다

요금부족판단 : 계산된 금액과 잔액을 비교해서 요금이 충분한지 판단해주는 연산이다

충분 : 금액이 충분한 경우 시행하는 연산이다

부족 : 금액이 부족한 경우 시행하는 연산이다

환승여부판단 : 하차시에환승을 했는지 판단하는 연산이다

환승요금계산 : 하차시 환승을 하였을 경우 계산되는 금액이다

미환승요금계산 : 하차시 환승 안했을 경우 계산되는 금액이다

정산계산 : 지하철과 버스단말기로부터 받은 정보로 정산 공식에 따라 계산하는 부분이다

저장-전송 : 정산한 금액을 각 버스와 지하철회사로 전송하는(저장하는) 부분이다

목차 항목을 찾을 수 없습니다.

미정산 미환승 : 지하철에서 이전에 지하철을 타고 미정산시 금액을 계산하는 부분이다 Definition, acronyms, and abbreviations

1.4 Reference

Hand-out

Overview

2 Overall Description

2.1 Product Perspective

대상 제품은 실제 교통 단말기와 교통카드에 사용될 수 있는 제품이 될 수 있다. HW (센서)에 의한 동작을 처리하고, 처리한 결과는 HW (화면)에 출력한다. 실제 HW 에 의한 동작은 SW 및 console 화면으로 처리하여 기능의 동작 유무를 확인하도록 한다.

2.2 Product functions

2.2.1 버스, 지하철 단말기

화면에 현재 시간을 표시한다. 승차시 부과 되어야 하는 요금을 계산 하여 교통카드내 잔액과 비교한 후 잔액이 부족할 시 부족 처리를 한다. 잔액이 충분할 시 화면에 요금을 출력 하고 교통카드와 단말기에 정보를 저장한다.

2.2.2 정산

교통카드 단말기에서 받아온 정보를 정산 공식을 적용하여 계산 한 후 버스 회사와 지하철 회사로 전송한다.

2.3 User characteristics

2.4 Constraints

현재 시간은 00 시 00 분 00 초 이다.

날짜의 표기법은 '월-일' 이다.

금액은 "₩"이다.

2.5 Assumptions and dependencies

버스 승차는 지역에 상관 없이 할 수 있다.

지하철은 2 호선 역 중 5 개만 고려한다: 건대입구, 동대문역사문화공원, 합정, 신림, 강남

하루는 3 분으로 가정한다.

다음 날 운행이 시작하기 전까지는 정산이 반드시 이루어 져야 한다.

정산후 모든 프로그램의 정보는 초기화 된다. (사용자 카드 정보 제외)

환승이 가능한 시간은 15 초 이내다.

버스 환승 시 추가요금의 기준인 단위 시간은 30 초이다.

버스와 지하철의 기본료는 1050 원이다.

교통카드는 저장된 텍스트 파일로 가정하고, 교통카드 태그 행위를 해당 파일을 입력

하는 것으로 가정한다.

잔액이 모자를 경우 태우지 않는다.

버스 환승 최고 부과금액인 700 원이 남아 있지 않으면 버스로

환승시키지 않는다.

지하철 환승 최고 부과금액인 600 원이 남아 있지 않으면 지하철로

환승시키지 않는다.

하루(3 분) 종료 시 탑승되어 있는 승객은 미정산으로 처리한다. (미정산 금액: 표 1,2 참

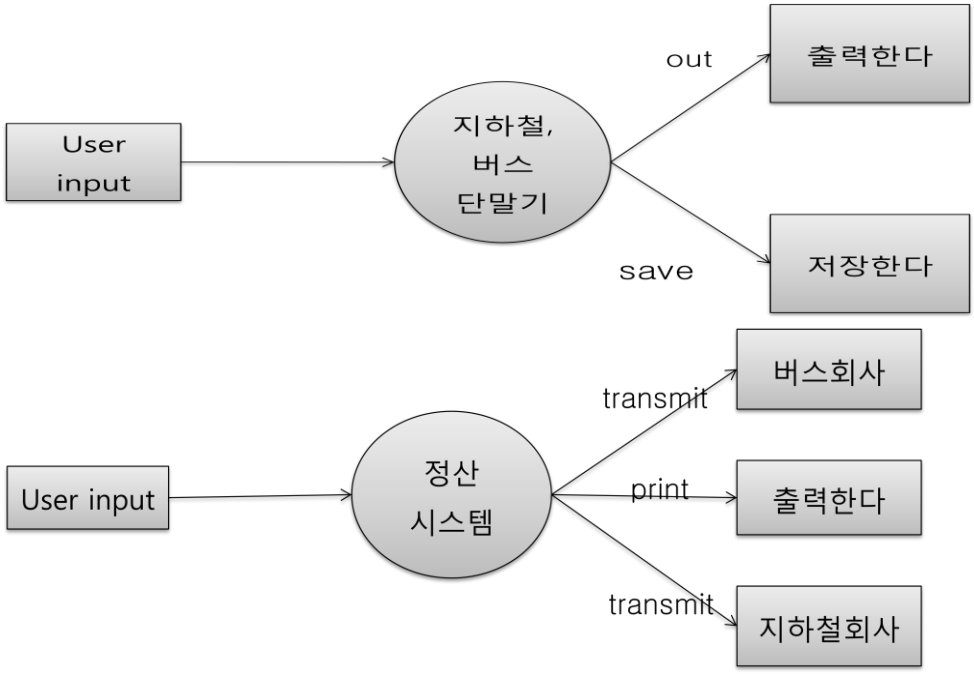
조)

정산시 소수점 이하는 반올림한다.

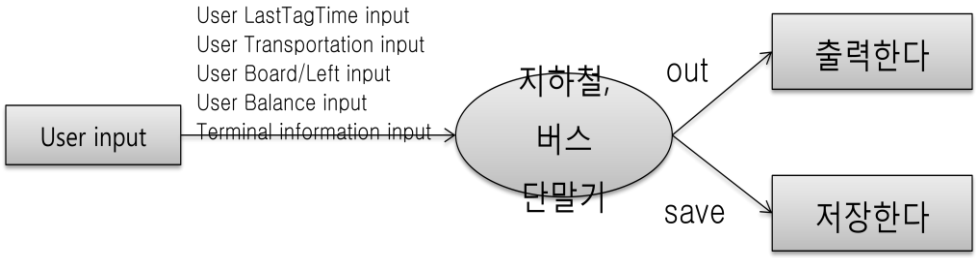
3 Structured Analysis

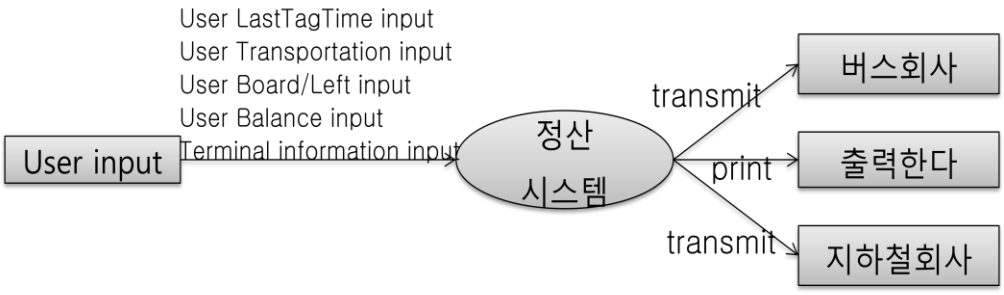
3.1 System Context Diagram

3.1.1 Basic System Context Diagram



3.1.2 The System Context Diagram

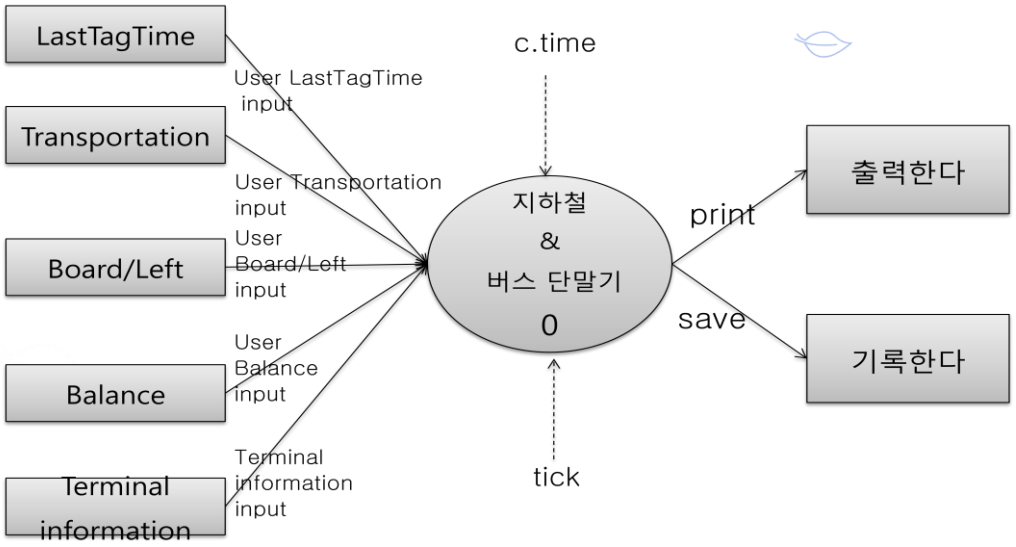




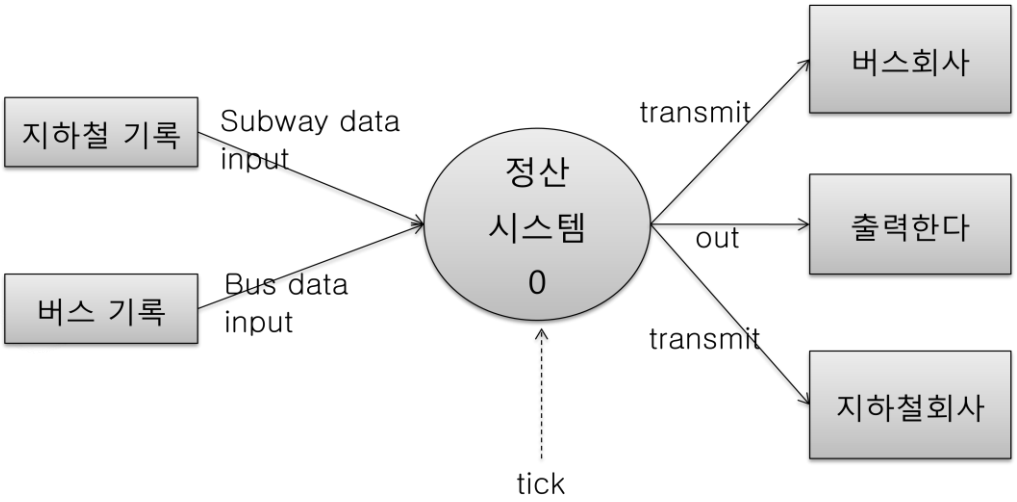
3.2 Data Flow Diagram

3.2.1 DFD level 0

3.2.1.1.DFD(버스&지하철)



3.2.1.2.DFD(정산)



3.2.1.3 Process specification

3.2.1.3.1 Data Dictionary(버스&지하철)

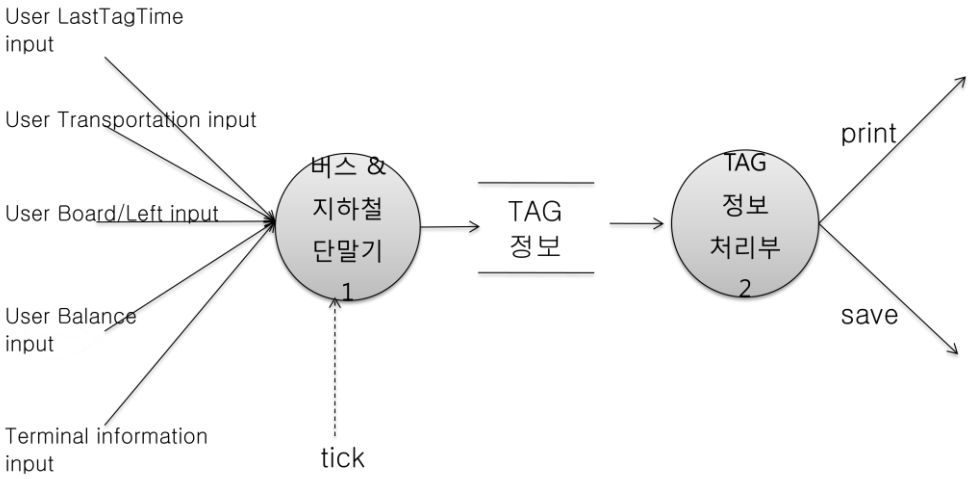
Input/Output Event	Description
User LastTagTime input	사용자 카드에서 마지막으로 태그된 시간 정보
User Transportation input	사용자 카드에 저장된 마지막으로 태그된 교통수단 정보
User Board/Left input	사용자 카드에 저장된 마지막으로 태그된 승하차 정보
User Balance input	사용자 카드에 저장된 잔액 정보
Terminal information input	사용자 카드에 저장된 마지막으로 태그된 단말기 정보
Out	태그된 시간, 결제금액 정보
Save	태그된 시간, 단말기의 교통수단, 승차하차 여부, 결제된 금액, 단말기의 번호정보

3.2.1.3.2 Data Dictionary(정산)

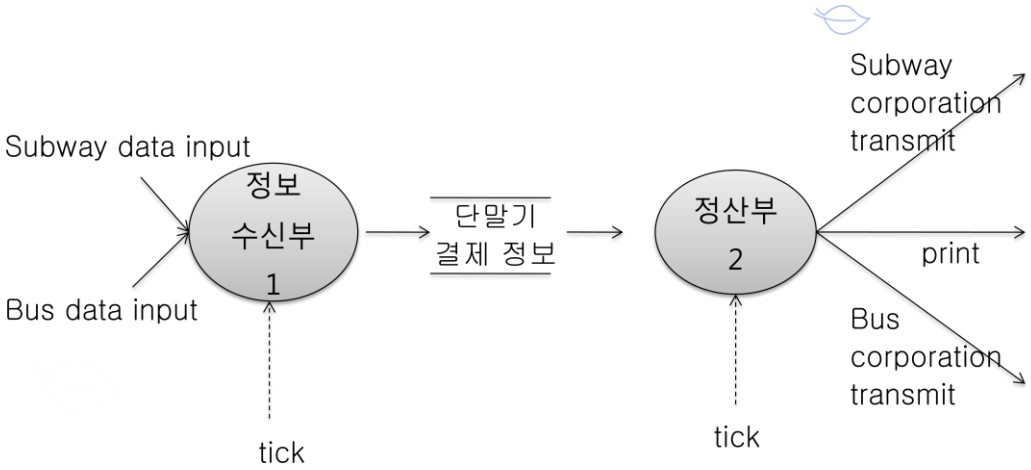
Input/Output Event	Description
User LastTagTime input	사용자 카드에서 마지막으로 태그된 시간 정보
User Transportation input	사용자 카드에 저장된 마지막으로 태그된 교통수단 정보
User Board/Left input	사용자 카드에 저장된 마지막으로 태그된 승하차 정보
User Balance input	사용자 카드에 저장된 잔액 정보
Terminal information input	사용자 카드에 저장된 마지막으로 태그된 단말기 정보
Transmit	버스/지하철에 대한 정산결과
Out	정산된 금액 모니터에 출력

3.2.2 DFD Level 1

3.2.2.1 DFD1(버스&지하철)

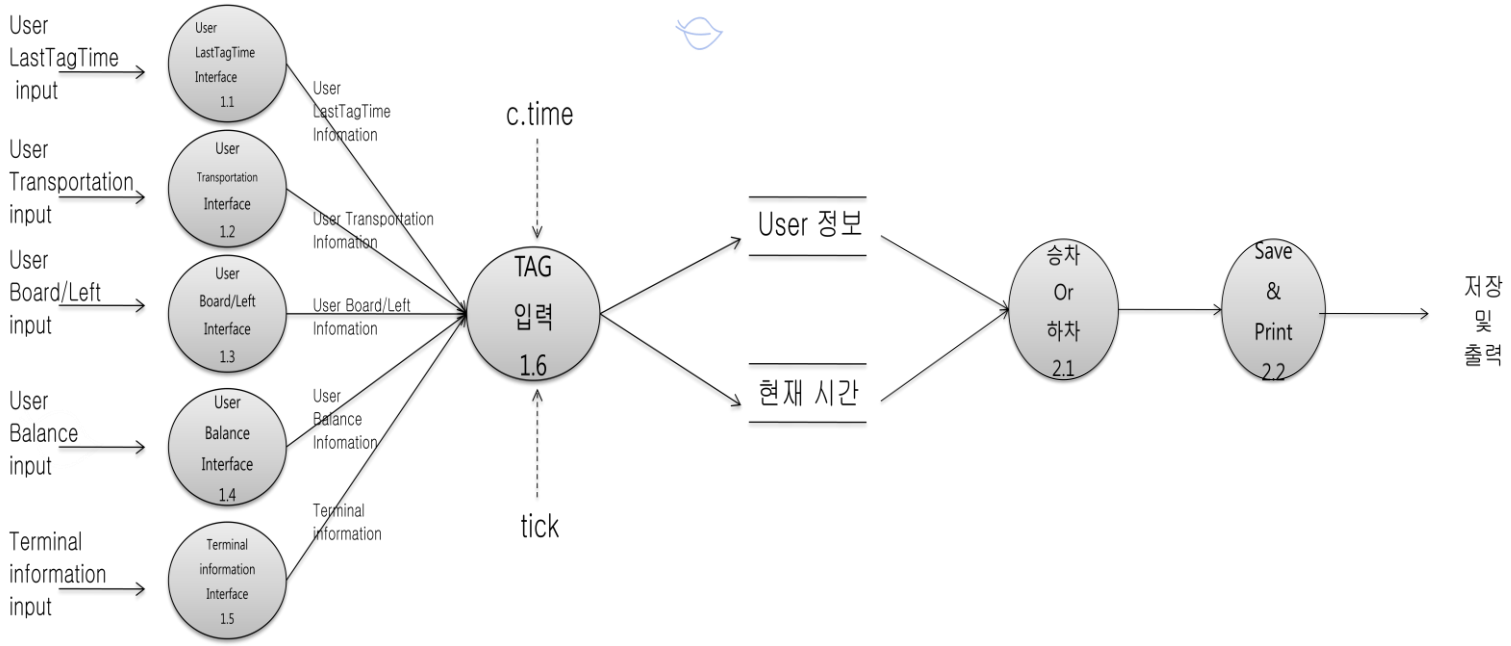


3.2.2.2 DFD1(정산)

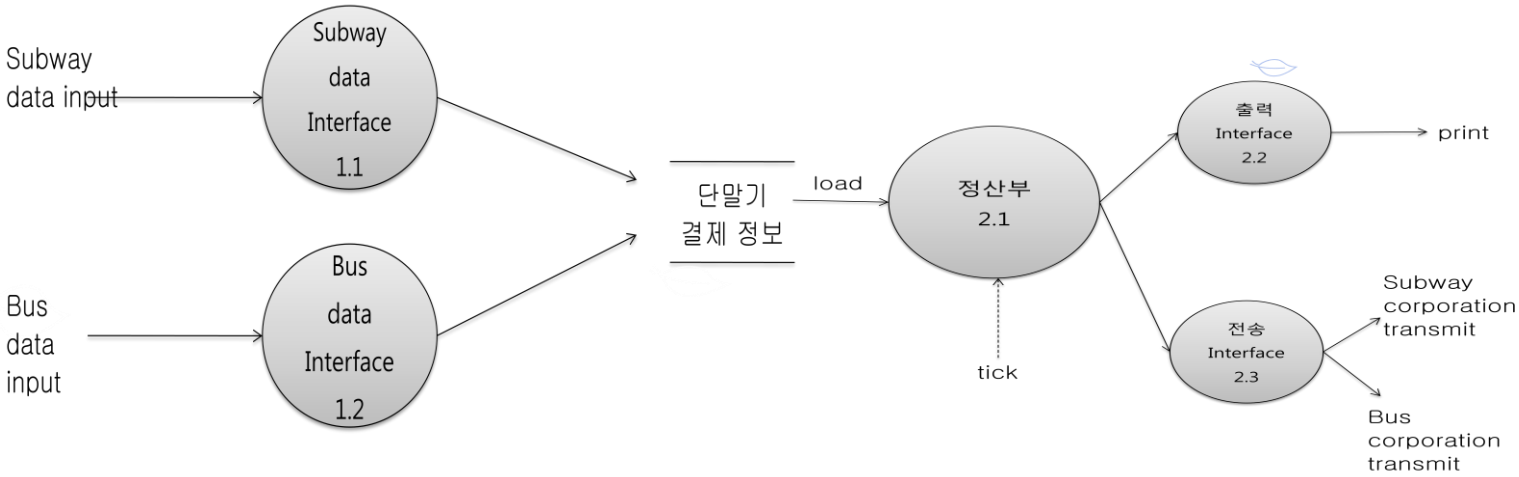


3.2.3 DFD Level 2

3.2.3.1 DFD2(버스&지하철)



3.2.3.2 DFD2(정산)



3.2.3.3 Process Specification(버스&지하철)

3.2.3.3.1 Process 1.1

Reference No.	1.1
Name	User LastTagTime Interface
Input	User LastTagTime input
Output	User LastTagTime Information
Process Description	받은 사용자 카드에서 마지막으로 태그된 시간 정보를 전달한다.

3.2.3.3.2 Process 1.2

Reference No.	1.2
Name	User Transportation Interface
Input	User Transportation input
Output	User Transportation Information
Process Description	받은 사용자 카드에 저장된 마지막으로 태그된 교통수단 정보를 전달한다.

3.2.3.3.3 Process 1.3

Reference No.	1.3
Name	User Board/Left Interface
Input	User Board/Left input
Output	User Board/Left Information
Process Description	받은 사용자 카드에 저장된 마지막으로 태그된 승하차 정보를 전달한다.

3.2.3.3.4 Process 1.4

Reference No.	1.4
Name	User Balance Interface
Input	User Balance input
Output	User Balance Information
Process Description	받은 사용자 카드에 저장된 잔액 정보를 전달한다.

3.2.3.3.5 Process 1.5

Reference No.	1.5
Name	Terminal information Interface
Input	Terminal information input
Output	Terminal information
Process Description	받은 사용자 카드에 저장된 마지막으로 태그된 단말기 정보를 전달한다.

3.2.3.4 Process Specification(정산)

3.2.3.4.1 Process 1.1

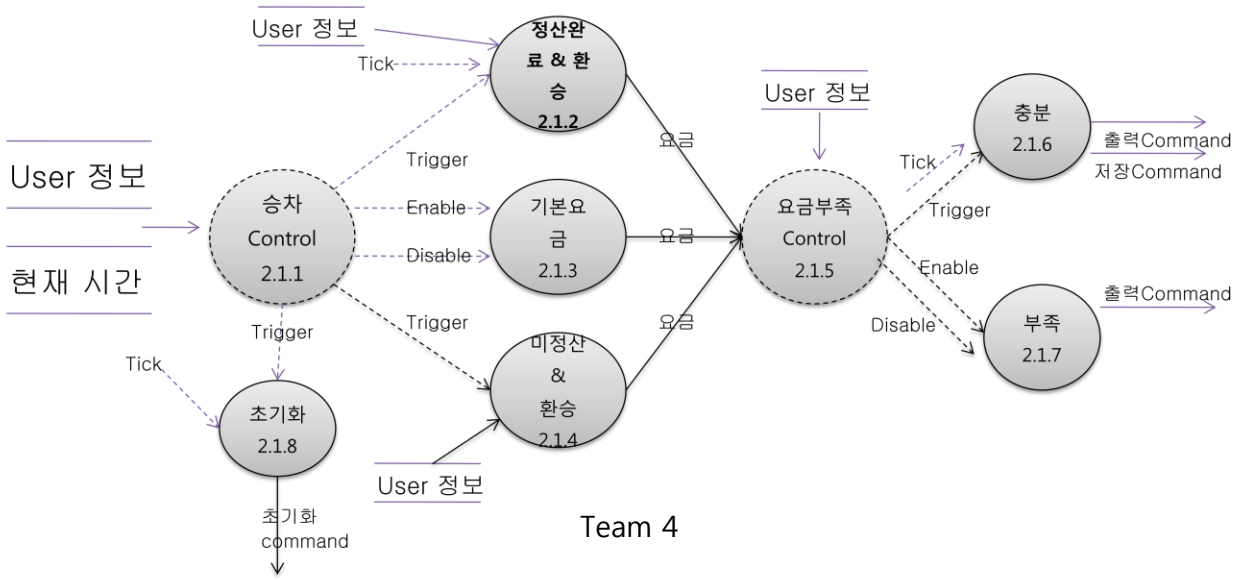
번호	정산 1.1
이름	Subway data Interface
입력	Subway data Input
출력	Subway data Information
설명	지하철의 데이터를 입력받아서 정보를 전송해 준다.

3.2.3.4.2 Process 1.2

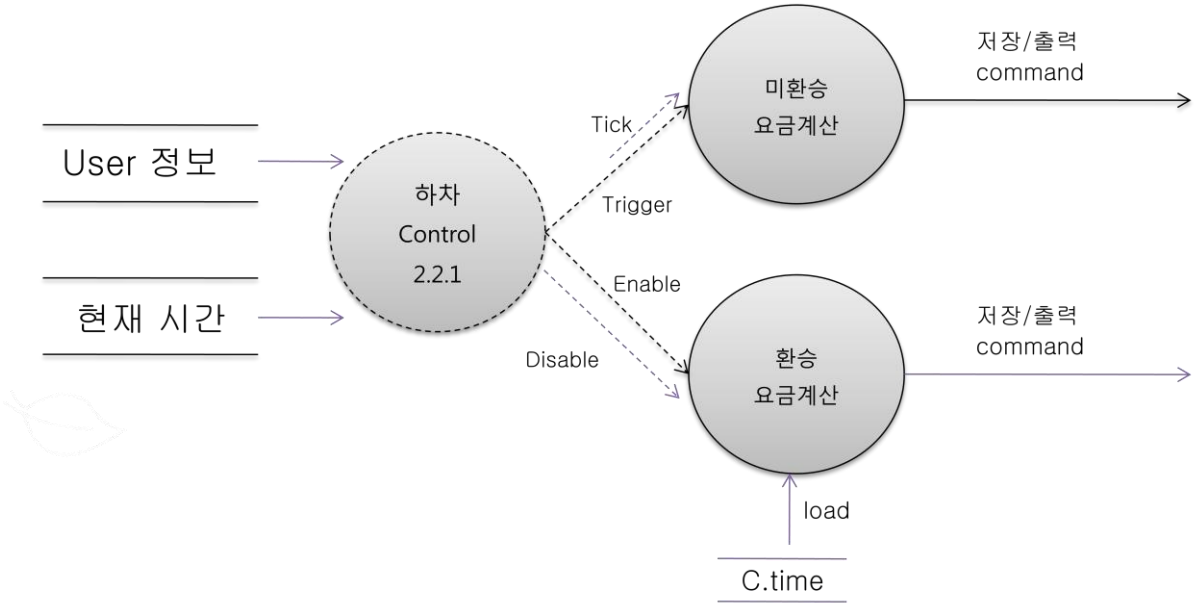
번호	정산 1.2
이름	Bus data Interface
입력	Bus data Input
출력	Bus data Information
설명	버스의 데이터를 입력받아서 정보를 전송해 준다.

3.2.4 DFD Level 3

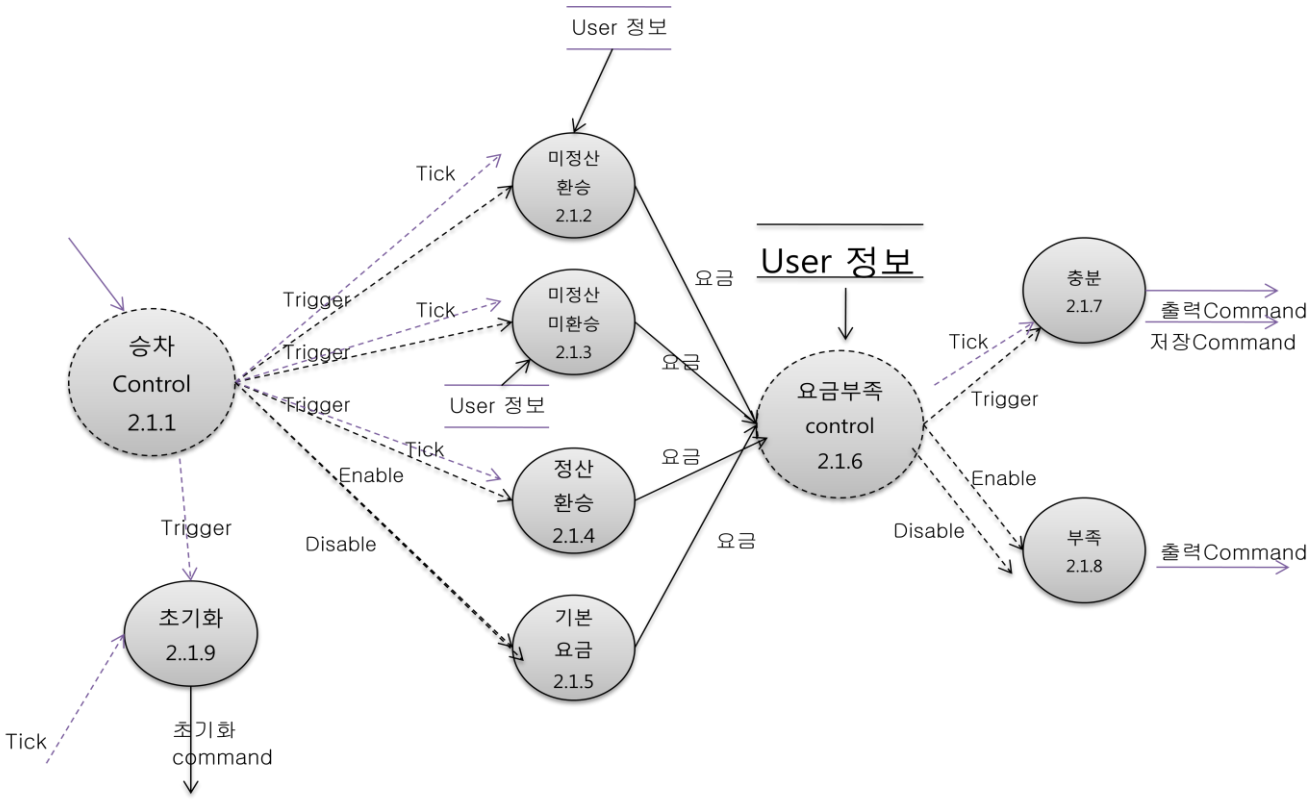
3.2.4.1 DFD3 버스-승차



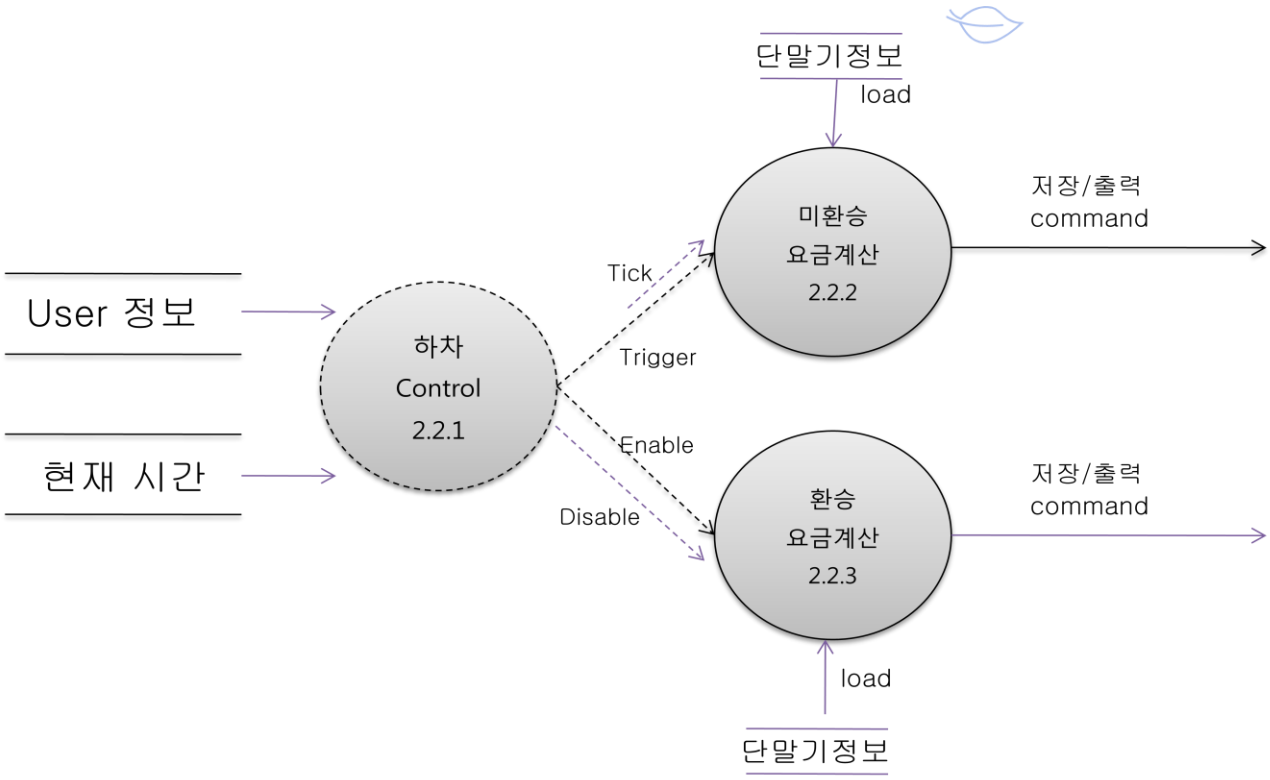
3.2.4.2 DFD3 버스-하차



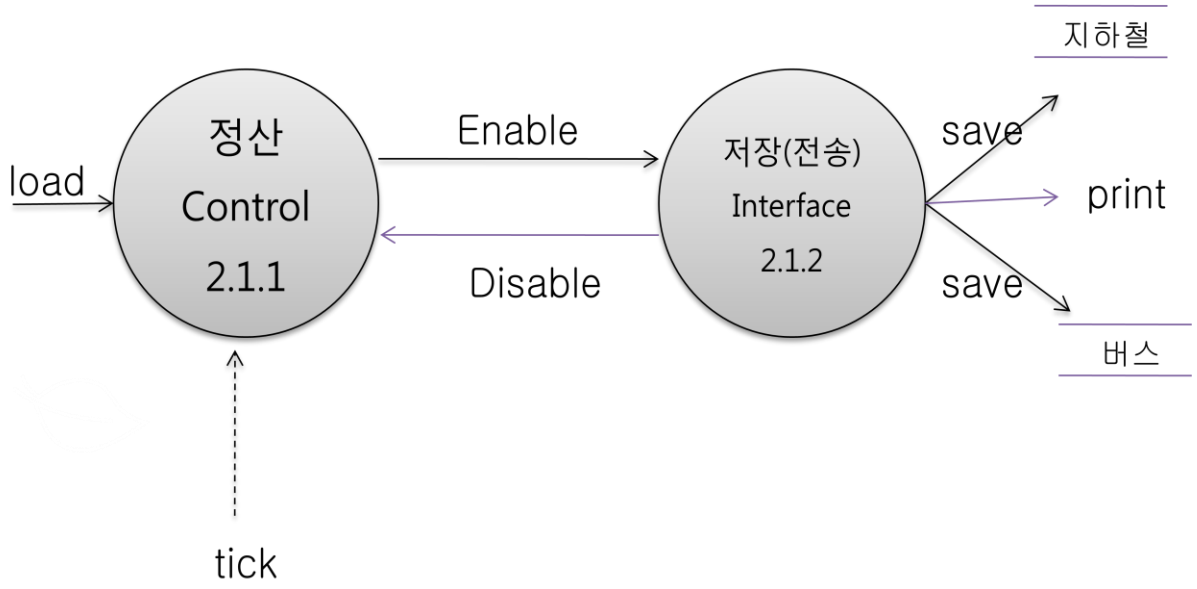
3.2.4.3 지하철-승차



3.2.4.4 지하철-하차



3.2.4.5 정산



3.2.4.6 Process Specification(버스)

3.2.4.6.1 Process 2.1.1

번호	버스 2.1.1
이름	승차 control
입력	User 정보, 현재시간
출력	trigger, enable, disable
설명	실질적으로 요금의 경우를 나누어 주는 주 컨트롤러로, 사용자의 정보를 입력받고, 환승여부, 미정산여부, 최초탑승여부의 조합에 따라 경우를 나누어 각 계산해야되는 요금 식을 판단해주고 다음 프로세스에 트리거를 전송해준다.

3.2.4.6.2 Process 2.1.2

번호	버스 2.1.2
이름	정산완료/환승
입력	User 정보, tick, trigger
출력	요금
설명	사용자가 이전에 정산을 하고 환승을 하였을 경우의 금액을 계산하고, 최대금액을 계산한 후 해당 금액들을 다음 컨트롤러로 전달해준다.

3.2.4.6.3 Process2.1.3

번호	버스 2.1.3
이름	기본요금
입력	enable, disable
출력	요금
설명	사용자가 최초탑승이거나 이전에 버스를 탔거나 환승시간이 넘은 경우, 기본금액을 다음 컨트롤러로 전달해준다.

3.2.4.6.4 Process2.1.4

번호	버스 2.1.4
이름	미정산/환승
입력	tick, trigger
출력	요금
설명	사용자가 미정산을하고 환승을 안한 경우의 금액을 계산하고, 최대금액을 계산한 후 해당 금액들을 다음 컨트롤러로 전달해준다.

3.2.4.6.5 Process2.1.5

번호	버스 2.1.5
이름	요금부족 Control

입력	요금
출력	enable, disable, trigger
설명	전달받은 금액을 유저의 잔액과 비교해서 요금부족 여부를 판단하여 각각 트리거를 전달해준다.

3.2.4.6.6 Process2.1.6

번호	버스 2.1.6
이름	부족
입력	enable disable
출력	출력 command
설명	최종처리에서 금액이 부족한 경우 활성화 되며, 금액 부족 문구를 출력하는 명령을 전송한다.

3.2.4.6.7 Process2.1.7

번호	버스 2.1.7
이름	충분
입력	Trigger
출력	저장 command, 출력 command
설명	요금정보와 그에 해당하는 정보를 사용자 카드와 단말기에 각각 저장시키고, 금액과 시간을 출력하는 명령을 전송한다.

3.2.4.6.8 Process2..2.1

번호	버스 2.2.1

이름	하차 Control
입력	User 정보, 현재 시간
출력	Trigger
설명	사용자가 하차태그를 하였을 시 환승을 하였는지 안하였는지의 여부에 대한 트리거를 다음 프로세스에 전송해준다.

3.2.4.6.9 Process2.2.2

번호	버스 2.2.2
이름	미환승 요금계산
입력	Trigger, Tick
출력	저장/출력 command
설명	하차시 미환승 하였을 경우 지금까지 이동한 시간정보를 입력받아서 해당하는 금액을 계산하고 해당 금액과 해당하는 정보를 사용자카드와 단말기에 각각 저장시키고, 금액과 시간을 출력한다.

3.2.4.6.10 Process2.2.3

번호	버스 2.2.3
이름	환승 요금계산
입력	Enable, Disable
출력	저장/출력 command
설명	하차시 환승 하였을 경우 지금까지 이동한 시간정보를 입력받아서 해당하는 금액을 계산하고 해당 금액과 해당하는

	정보를 사용자카드와 단말기에 각각 저장시키고, 금액과 시간을 출력한다.

3.2.4.7 Process Specification(지하철)

3.2.4.7.1 Process2.1.1

번호	지하철 2.1.1
이름	승차 control
입력	User 정보, 현재시간
출력	trigger, enable, disable
설명	실질적으로 요금의 경우를 나누어 주는 주 컨트롤러로, 사용자의 정보를 입력받고, 환승여부, 미정산여부, 최초탑승여부의 조합에 따라 경우를 나누어 각 계산해야되는 요금 식을 판단해주고 다음 프로세스에 트리거를 전송해준다.

3.2.4.7.2 Process2.1.2

번호	지하철 2.1.2
이름	미정산 환승
입력	tick, trigger
출력	요금
설명	사용자가 이전에 미정산을 하고 환승을 하였을 경우의 금액을 계산하고, 최대금액을 계산한 후 해당 금액들을 다음 컨트롤러로 전달해준다.

3.2.4.7.3 Process2.1.3

번호	지하철 2.1.3
이름	미정산 미환승
입력	tick, trigger
출력	요금
설명	사용자가 미정산을하고 환승을 안한 경우의 금액을 계산하고, 최대금액을 계산한 후 해당 금액들을 다음 컨트롤러로 전달해준다.

3.2.4.7.4 Process2.1.4

번호	지하철 2.1.4
이름	정산/환승
입력	tick, trigger
출력	요금
설명	사용자가 정산을하고 환승을 안한 경우의 금액을 계산하고, 최대금액을 계산한 후 해당 금액들을 다음 컨트롤러로 전달해준다.

3.2.4.7.5 Process2.1.5

번호	지하철 2.1.5
이름	기본요금
입력	enable, disable
출력	요금
설명	사용자가 최초탑승이거나 이전에 버스를 탔거나 환승시간이 넘은 경우, 기본금액을 다음 컨트롤러로 전달해준다.

3.2.4.7.6 Proces21.6

번호	지하철 2.1.6
이름	요금부족 Control
입력	요금, User 정보
출력	enable, disable, trigger
설명	전달받은 금액을 유저의 잔액과 비교해서 요금부족 여부를 판단하여 각각 트리거를 전달해준다.

3.2.4.7.7 Process2.1.7

번호	지하철 2.1.7
이름	충분

입력	Trigger
출력	저장command, 출력 command
설명	요금정보와 그에 해당하는 정보를 사용자 카드와 단말기에 각각 저장시키고, 금액과 시간을 출력하는 명령을 전송한다.

3.2.4.7.8 Process2.1.8

번호	지하철 2.1.8
이름	부족
입력	Enable, Disable
출력	출력 command
설명	금액이 부족한 경우 활성화 되며, 금액 부족 문구를 출력하는 명령을 전송한다.

3.2.4.7.9 Process2.1.9

번호	지하철 2.1.9
이름	초기화 Trigger
입력	Trigger, Tick
출력	초기화 command
설명	3분이 되었을 시 초기화를 시켜주는 부분이다.

3.2.4.7.10 Process2.2.1

번호	지하철 2.2.1
이름	하차 Control
입력	User 정보, 현재 시간
출력	Trigger
설명	사용자가 하차태그를 하였을 시 환승을 하였는지 안하였는지의 여부에 대한 트리거를 다음 프로세스에 전송해준다.

3.2.4.7.11 Process2.2.2

번호	지하철 2.2.2

이름	미환승 요금계산
입력	Trigger, Tick
출력	저장/출력 command
설명	하차시 미환승 하였을 경우 지금까지 이동한 정거장 정보를 입력받아서 해당하는 금액을 계산하고 해당 금액과 해당하는 정보를 사용자카드와 단말기에 각각 저장시키고, 금액과 시간을 출력한다.

3.2.4.7.12 Process2.2.3

번호	지하철 2.2.3
이름	환승 요금계산
입력	Enable, Disable
출력	저장/출력 command
설명	하차시 환승 하였을 경우 지금까지 이동한 정거장 정보를 입력받아서 해당하는 금액을 계산하고 해당 금액과 해당하는 정보를 사용자카드와 단말기에 각각 저장시키고, 금액과 시간을 출력한다.

3.2.4.8 Process Specification(지하철)

3.2.4.8.1 Process2.1.1

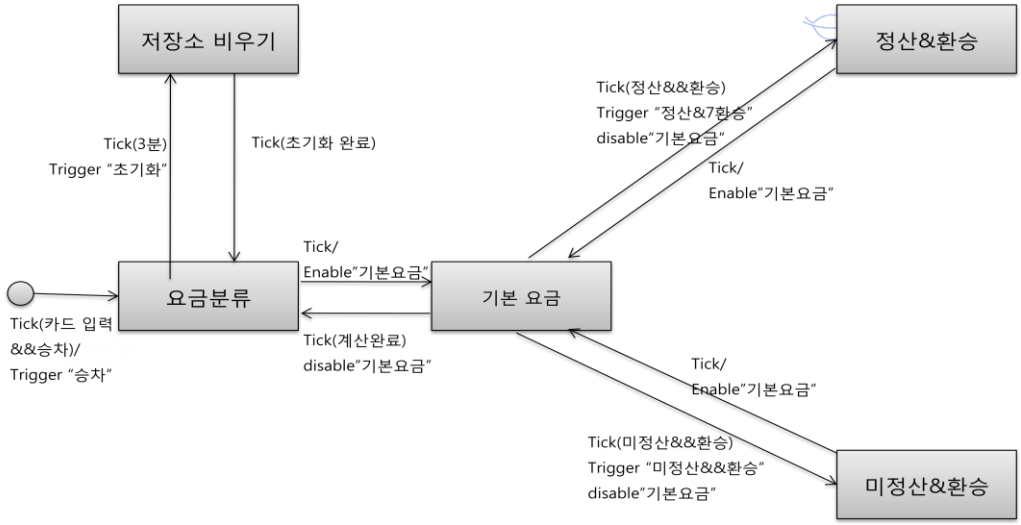
번호	정산 2.1.1
이름	정산 control
입력	load
출력	enable, disable
설명	각 환승방향별로 정산금액을 계산해주고, 해당 정보를 저장하며, 3분마다 전송부를 활성화 시킨다.

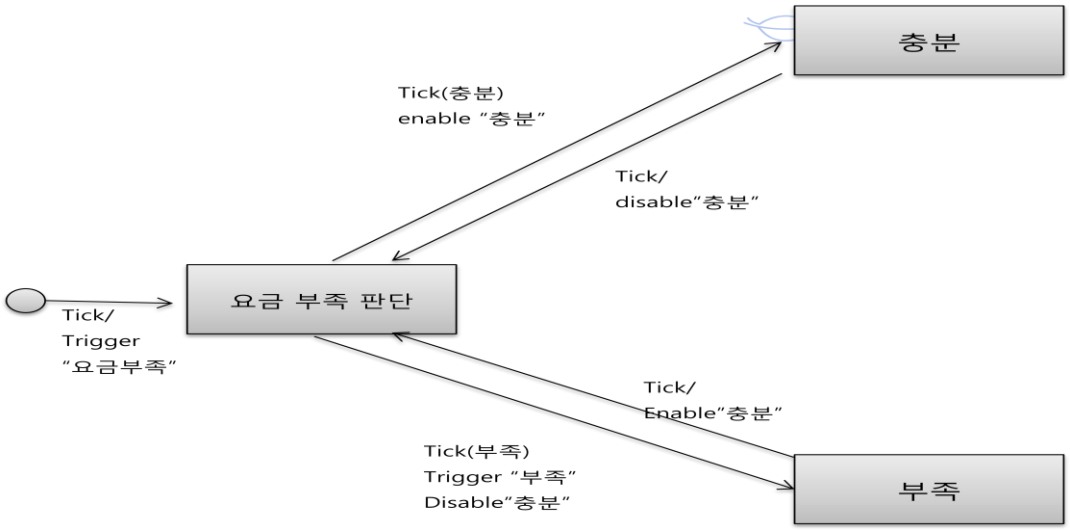
3.2.4.8.2 Process2.1.2

번호	정산 2.1.2
이름	정산 control
입력	enable, disable
출력	save, print
설명	정산된 금액에 대해서 각각 지하철과 버스회사로 전송(저장)해주고, 정산된 금액을 출력해준다.

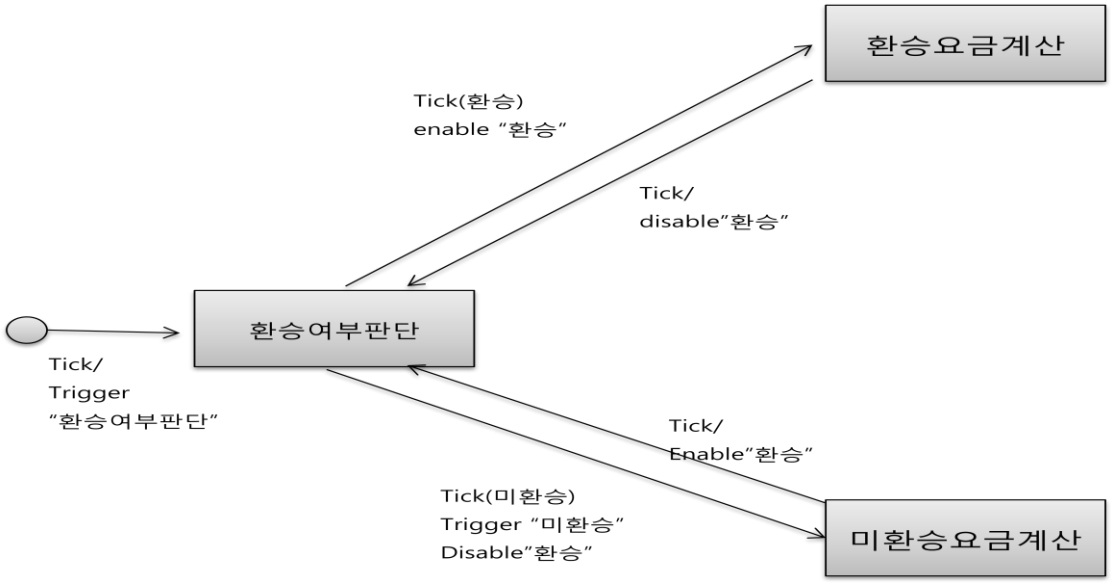
3.2.5 DFD Level 4

3.2.5.1 DFD4(버스)-승차

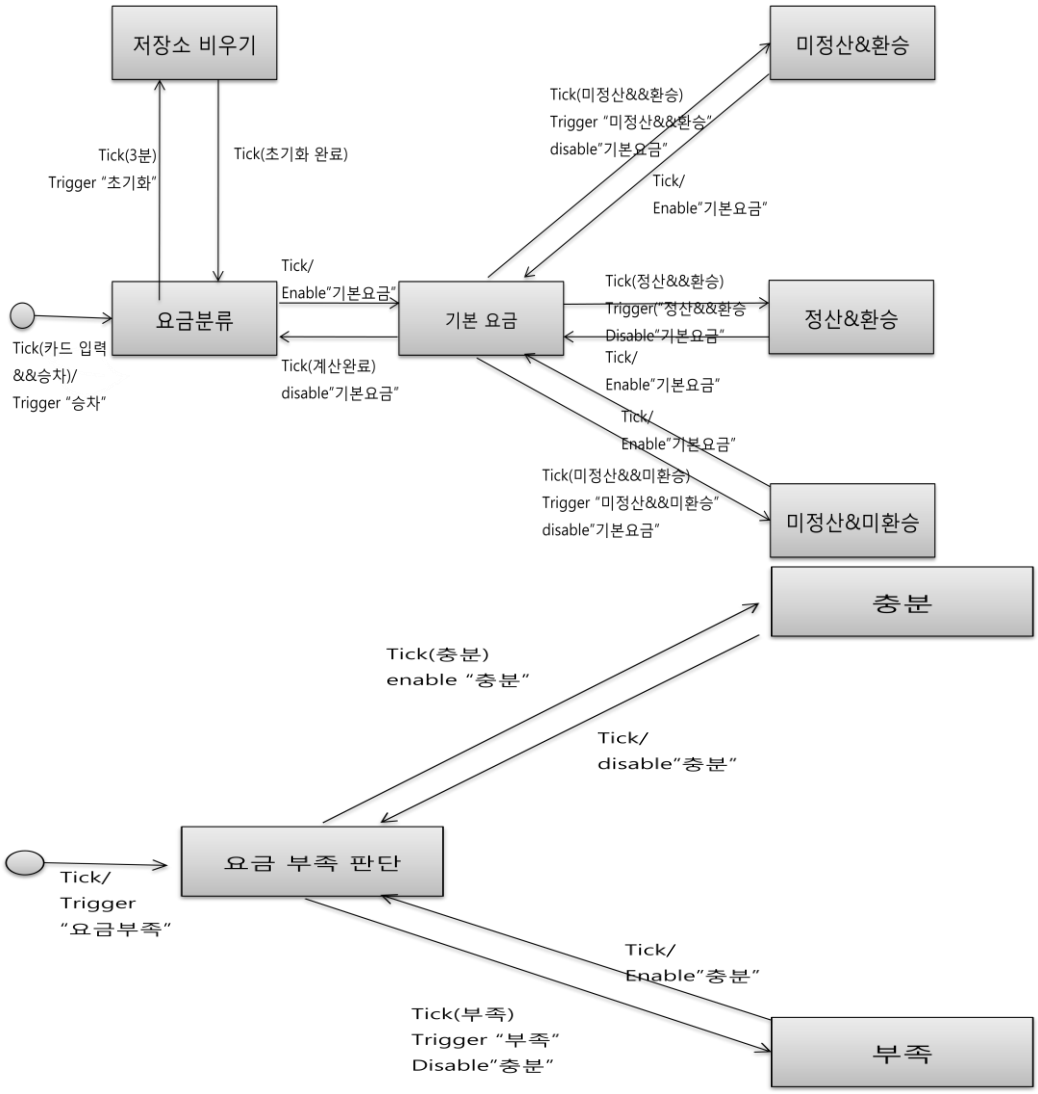




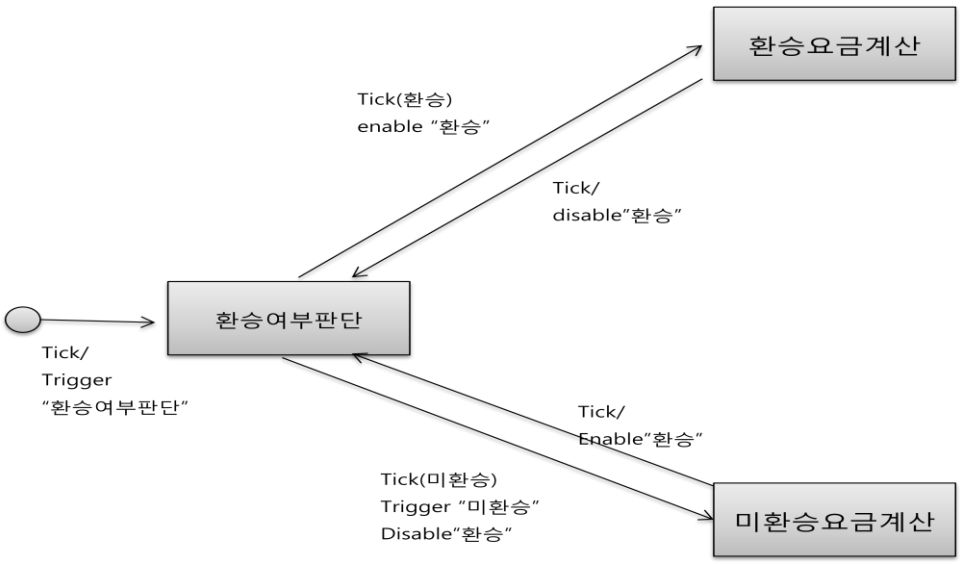
3.2.5.2 DFD(버스)-하차



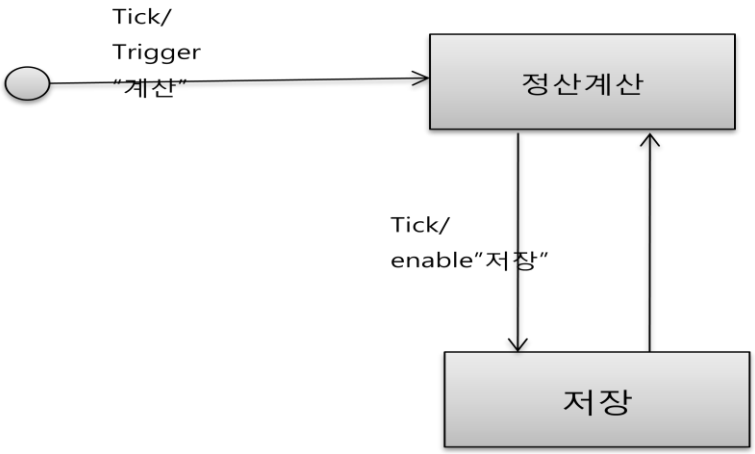
3.2.5.3 DFD4(지하철)-승차



3.2.5.4 DFD(지하철)-하차

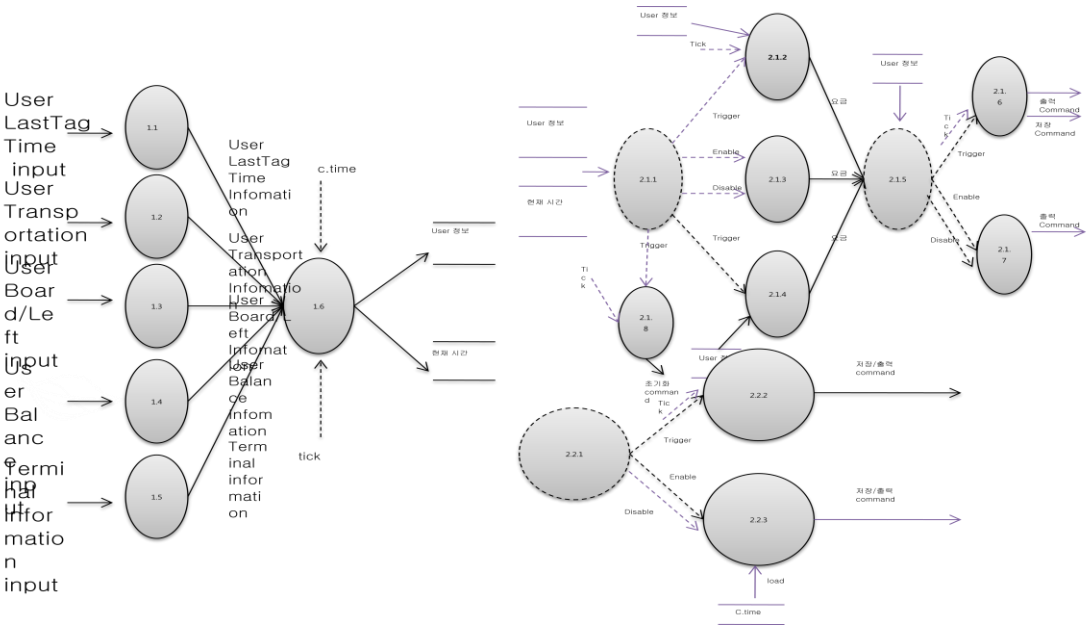


3.2.5.5 DFD4(정산)

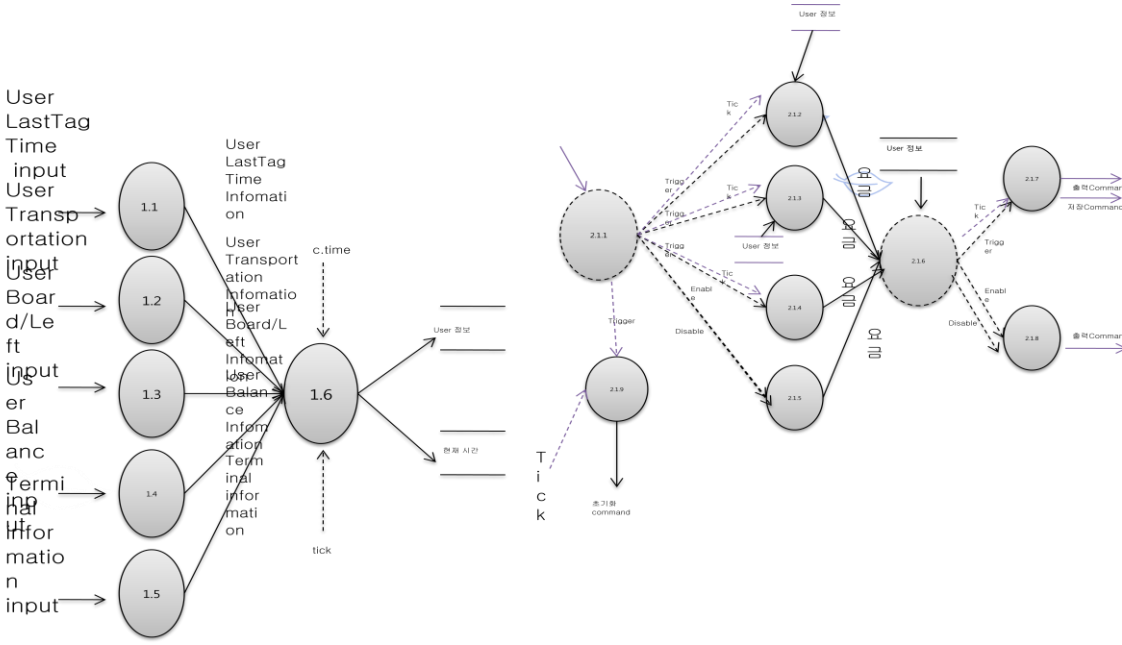
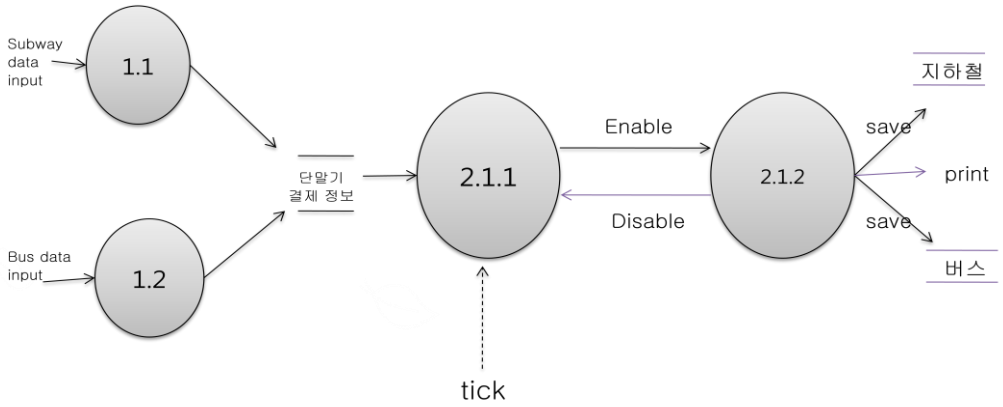


3.2.6 Overall DFD

3.2.6.1 Overall DFD(버스)



3.2.6.2 Overall DFD(지하철)



3.2.6.3 Overall DFD(정산)