

# Software Requirement Analysis for Coffee Machine System

Project Team

**Team 1**

Date

**2016-09-25**

---

**Team Information**

**201411259 교수창**

**201511243 김동언**

**201511263 박종엽**

**201511280 이선엽**

## Table of Contents

1	Introduction _____	4
1.1	Purpose _____	4
1.2	Scope _____	4
1.3	Definition, acronyms, and abbreviations _____	4
1.4	Reference _____	4
1.5	Overview _____	4
2	Overall Description _____	4
2.1	Product Perspective _____	4
2.2	Product functions _____	4
2.3	User characteristics _____	5
2.4	Constraints _____	5
2.5	Assumptions and dependencies _____	5
3	Structured Analysis _____	5
3.1	System Context Diagram _____	5
3.1.1	Basic System Context Diagram _____	5
3.1.2	Event List _____	5
3.1.3	The System Context Diagram _____	6
3.2	Data Flow Diagram _____	6
3.2.1	DFD level 0 _____	6
3.2.1.1	DFD _____	6
3.2.1.2	Process Specification _____	7
3.2.1.2.1	Process 1 _____	7
3.2.1.2.2	... _____ 오류! 책갈피가 정의되어 있지 않습니다.	
3.2.1.2.3	Process # _____ 오류! 책갈피가 정의되어 있지 않습니다.	
3.2.1.3	Data Dictionary _____	8
	Team 1	2

3.2.2	DFD Level # _____	9
3.2.2.1	DFD _____	9
3.2.2.2	Process Specification _____	10
3.2.2.2.1	Process #.1 _____	10
3.2.2.2.2	... _____ 오류! 책갈피가 정의되어 있지 않습니다.	
3.2.2.2.3	Process #.# _____ 오류! 책갈피가 정의되어 있지 않습니다.	
3.2.2.3	Data Dictionary _____	11
3.2.2.4	State Transition Diagram ( <i>Name of Controller</i> ) _____	21
3.2.3	Overall DFD _____	25

## 1 Introduction

### 1.1 Purpose

본 문서는 2016년 건국대학교의 소프트웨어공학 개론 강의의 실습과제를 설명한다. 실습 과제는 coffee machine을 소프트웨어로 구성된 가상의 시스템으로 구현하는 것이다.

### 1.2 Scope

Coffee Machine은 사용자의 요청에 따라 machine의 상태를 체크한 후 커피를 추출한다. 2016SE 수업을 통해 진행하게 될 프로젝트는 coffee machine을 가상 시스템으로 구현하는 것으로, coffee machine은 자신의 상태를 관리하여 사용자에게 알려주고, 사용자로부터 명령 및 입력 내용을 받아 출력을 만들어내는 역할을 한다. 모든 시스템은 SW 만으로 구현한다. HW가 필요한 부분은 SW 모듈을 만들어 가상의 HW를 구현한다.

### 1.3 Definition, acronyms, and abbreviations

HW : Hardware

SW : Software

CM : Coffee Machine

데이터 Format, Type은 C Style을 따른다.

### 1.4 Reference

None

### 1.5 Overview

2장 개발 대상에 대한 설명

3장 세부 기능 명세

## 2 Overall Description

### 2.1 Product Perspective

SW로 개발하는 가상의 Coffee machine 시스템

### 2.2 Product functions

Coffee machine은 커피 추출이 가능하다.

Coffee machine의 커피 추출은 조건에 따라 불가능 할 수 있다.

Coffee machine의 커피 추출은 사용자의 기호에 따라 세팅 할 수 있다.

Coffee machine은 예약이 가능하다.

Coffee machine은 커피가루가 없을 경우 원두를 자동으로 분쇄 후 커피를 추출

Coffee machine은 물과 커피의 잔량을 지속적으로 사용자에게 알려 준다.  
 Coffee machine의 물과 커피는 사용자에게 의해 충전된다.  
 Coffee machine은 machine의 내부를 청소 할 수 있다.

2.3 User characteristics

사용자는 언제든지 전원을 끌 수 있다.

2.4 Constraints

모든 명령 및 입력은 Cygwin의 command로 대체된다.  
 모든 출력은 Cygwin의 화면과 경보음으로 대체된다.  
 비 휘발성 특성이 필요한 정보들은 파일을 통해 관리 한다.  
 파일 저장이 필요한 경우 아래 경로에 파일 이름을 설정하여 저장 한다.  
 (C:WWSEWWT1WW???.txt)

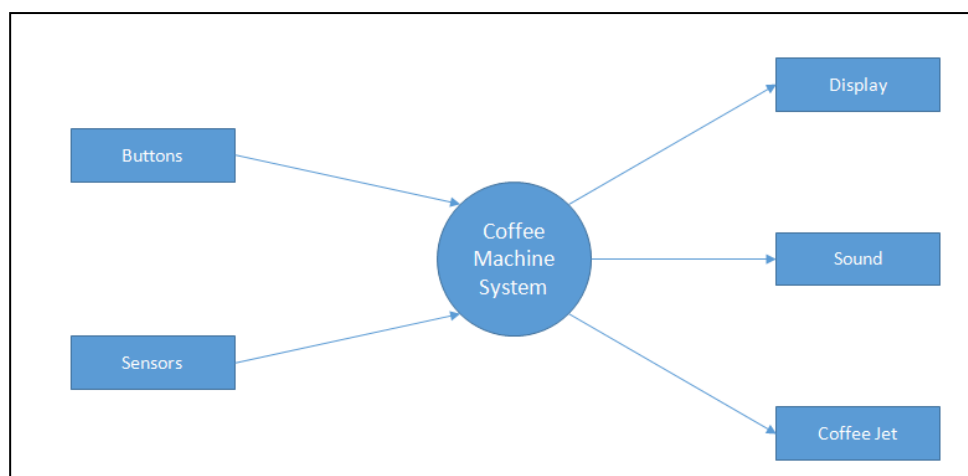
2.5 Assumptions and dependencies

청소와 커피 가루 제거는 다르다.  
 재료 투입 시 원두와 커피 가루는 독립적인 동작이다.  
 커피가 한번 추출되면 커피 가루가 남게 된다 (추후 커피 추출을 위해 이를 반드시 제거해야 한다.)

3 Structured Analysis

3.1 System Context Diagram

3.1.1 Basic System Context Diagram

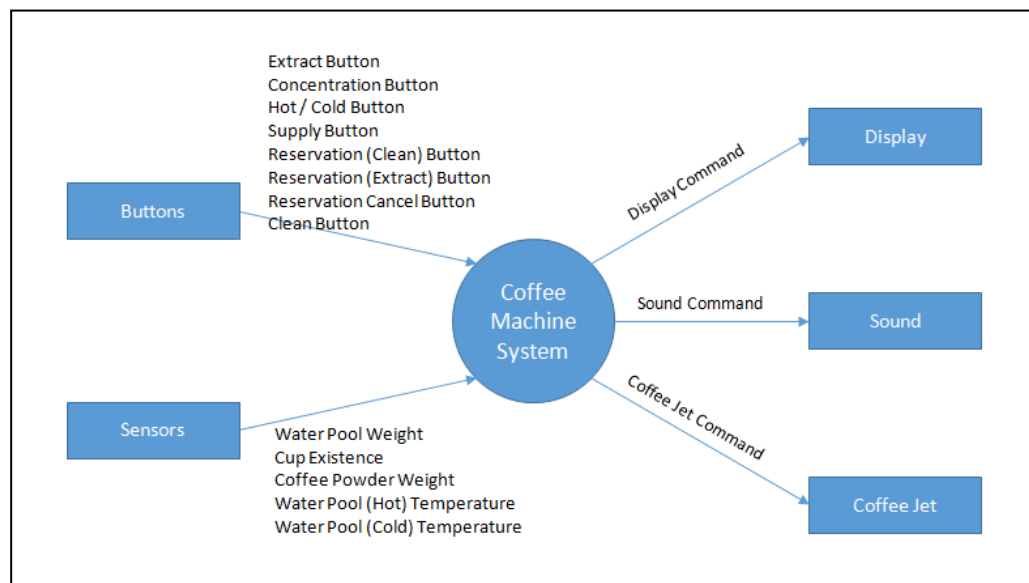


3.1.2 Event List

Input / Output Event	Description
----------------------	-------------

Extract Button	Enter into extraction mode
Supply Button	Enter into supplying mode
Hot / Cold Button	Sets temperature (Hot / Cold)
Concentration Button	Sets concentration (Weak / Normal / Strong)
Clean Button	Enter into cleaning mode
Reservation (Clean) Button	Enter into cleaning reservation mode
Reservation (Extract) Button	Enter into extracting reservation mode
Reservation Cancel Button	Cancel reservation
Water Pool Weight Sensor	Weight of water in common water pool
Cup Existence Sensor	Existence of cup under coffee jet
Coffee Powder Weight Sensor	Weight of coffee powder
Temperature (Hot) Sensor	Temperature of hot water
Temperature (Cold) Sensor	Temperature of cold water
Display Command	Displays string
Sound Command	Controls speaker
Coffee Jet Command	Controls coffee jet

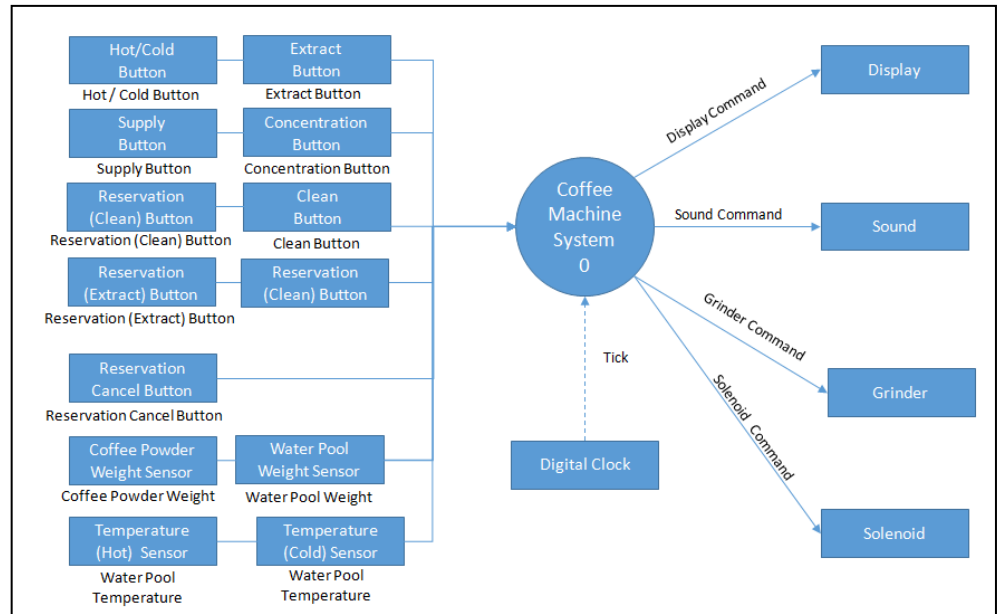
### 3.1.3 The System Context Diagram



## 3.2 Data Flow Diagram

### 3.2.1 DFD level 0

#### 3.2.1.1 DFD



### 3.2.1.2 Process Specification

#### 3.2.1.2.1 Process 0

<b>Reference No</b>	0
<b>Name</b>	Coffee Machine System
<b>Input</b>	Extract Button Concentration Button Clean Button Water Pool Weight Reservation Cancel Button Reservation Clean Button Reservation Extract Button Water Pool Temperature
<b>Output</b>	Display Command Sound Command Grinder Command Solenoid Command
<b>Process Description</b>	<ol style="list-style-type: none"> <li>1. 사용자가 Extract Button을 누르면 조건을 확인하여 커피를 추출함</li> <li>2. Reservation 류 버튼을 통해 예약동작 가능</li> <li>3. 물의 무게와 물의 온도를 입력 받아 제어</li> <li>4. 사용자에게 현재 CM 상태를 보이기 위</li> </ol>

	<p>해 Display Command를 전송</p> <p>5. 사용자에게 현재 상태를 소리로 전달하기 위해 Sound Command를 전송</p> <p>6. 원두를 갈기 위해 Grinder에 Grinder Command를 전송</p> <p>7. 물의 양과 원두의 양을 조절하기 위해 Solenoid Command를 전송</p>
--	--

3.2.1.3 Data Dictionary

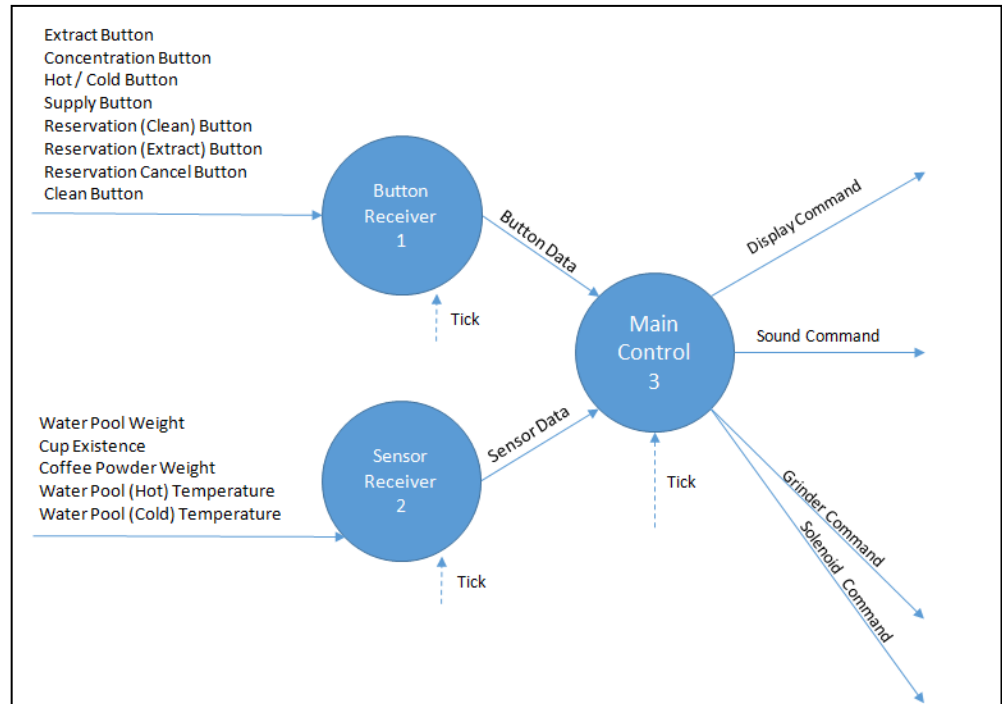
Input/ Output Event	Description	Format, Type
Extract Button	Enter into extraction mode	int is_pushed; #define RELEASE 0 #define PUSH 1
Supply Button	Enter into supplying mode	int is_pushed; #define RELEASE 0 #define PUSH 1
Hot / Cold Button	Sets temperature (Hot / Cold)	int is_pushed; #define RELEASE 0 #define PUSH 1
Concentration Button	Sets concentration (Weak / Normal / Strong)	int is_pushed; #define RELEASE 0 #define PUSH 1
Clean Button	Enter into cleaning mode	int is_pushed; #define RELEASE 0 #define PUSH 1
Reservation Button (Clean)	Enter into cleaning reservation mode	struct date{ int year; int month; int day; int time; }  int is_pushed; #define RELEASE 0 #define PUSH 1
Reservation Button (Extract)	Enter into extracting reservation mode	struct date{ int year; int month;



		<pre> int day; int time;  }  int is_pushed; #define RELEASE 0 #define PUSH 1 </pre>
Water Pool Weight Sensor	Weight of water in common water pool	float water_weight;
Cup Existence Sensor	Existence of cup under coffee jet	<pre> int is_cup_exist; #define NOT_EXIST 0 #define EXIST 1 </pre>
Coffee Powder Weight Sensor	Weight of coffee powder	float powder_weight
Temperature (Hot) Sensor	Temperature of hot water	float hot_water_temperature;
Temperature (Cold) Sensor	Temperature of cold water	float cold_water_temperature;
Display Command	Displays string	string display_string;
Sound Command	Controls speaker	<pre> bool beep; //beep-beep On / Off </pre>
Grinder Command	Controls Grinder	On / Off
Solenoid Command	Controls Solenoid	On / Off

### 3.2.2 DFD Level 1

#### 3.2.2.1 DFD



### 3.2.2.2 Process Specification

#### 3.2.2.2.1 Process 1

<b>Reference No</b>	1
<b>Name</b>	Button Receiver
<b>Input</b>	Extract Button Concentration Button Hot / Cold Button Supply Button Reservation (Clean) Button Reservation (Extract) Button Reservation Cancel Button Clean Button Tick
<b>Output</b>	Button Data
<b>Process Description</b>	1. 사용자가 입력한 버튼에 대해 데이터를 가공하여 Button Data형식으로 Main Control에 전달

#### 3.2.2.2.2 Process 2

<b>Reference No</b>	2
<b>Name</b>	Sensor Receiver

<b>Input</b>	Water Pool Weight Cup Existence Coffee Powder Weight Water Pool (Hot) Temperature Water Pool (Cold) Temperature Tick
<b>Output</b>	Sensor Data
<b>Process Description</b>	1. CM내부에서 커피를 추출하는데 필요한 요소들을 센서로 입력 받아 Sensor Data 형태로 Main Control에 전달

## 3.2.2.2.3 Process 3

<b>Reference No</b>	3
<b>Name</b>	Main Control
<b>Input</b>	Button Data Sensor Data Tick
<b>Output</b>	Display Command Sound Command Grinder Command Solenoid Command
<b>Process Description</b>	1. Button Receiver로부터 가공된 구조체 형태의 Button Data를 입력 받아 사용자가 지정한 동작을 수행 2. 사용자가 지정한 동작과 CM자체 동작을 수행하는데 있어 필요한 요소에 대한 정보는 Sensor Data로부터 얻음

## 3.2.2.3 Data Dictionary

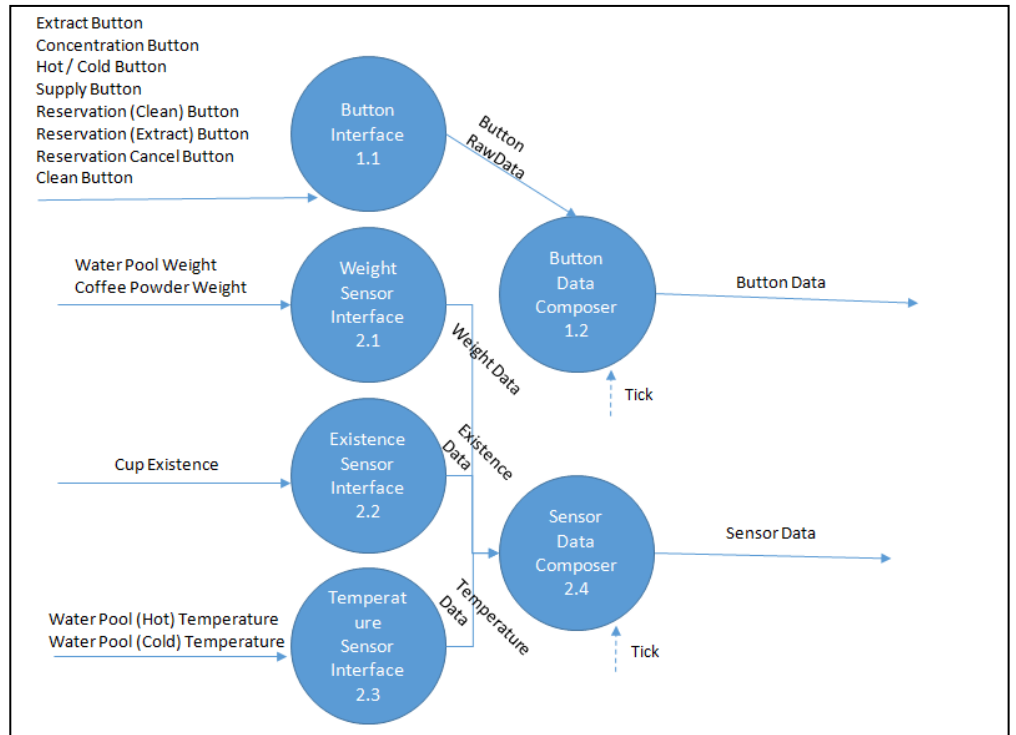
<b>Input/ Output Event</b>	<b>Description</b>	<b>Format, Type</b>
Button Data	Processed button data	Associative array Periodic
Button Data[Extract]	Is extract button pressed?	Boolean, Periodic
Button Data[Concentration]	Concentration (Weak / Normal / Strong)	Weak / Normal / Strong, Periodic
Button Data[Temperature]	Temperature (Hot / Cold)	Hot / Cold, Periodic
Button Data[Supply]	Is supply button	Boolean, Periodic

	pressed?	
Button Data[RClean]	Is reservation (clean) button pressed?	Boolean, Periodic
Button Data[RExtract]	Is reservation (extract) button pressed?	Boolean, Periodic
Button Data[RCancel]	Is reservation cancel button pressed?	Boolean, Periodic
Button Data[Clean]	Is clean button pressed	Boolean, Periodic
Button Data[Concentration]	Set Concentration	int concentration; #define STRONG 3 #define NORMAL 2 #define WEAK 1
Sensor Data	Processed sensor data	Associative array, Periodic
Sensor Data[Water]	Weight of common water pool	Float, Periodic
Sensor Data[Cup]	Existence of cup under coffee jet	Boolean, Periodic
Sensor Data[Powder]	Weight of coffee powder	Float, Periodic
Sensor Data[CTemp]	Temperature of cold water pool	Float, Periodic
Sensor Data[HTemp]	Temperature of hot water pool	Float, Periodic
Sensor Data[Bean]	Weight of coffee beans	float beans;

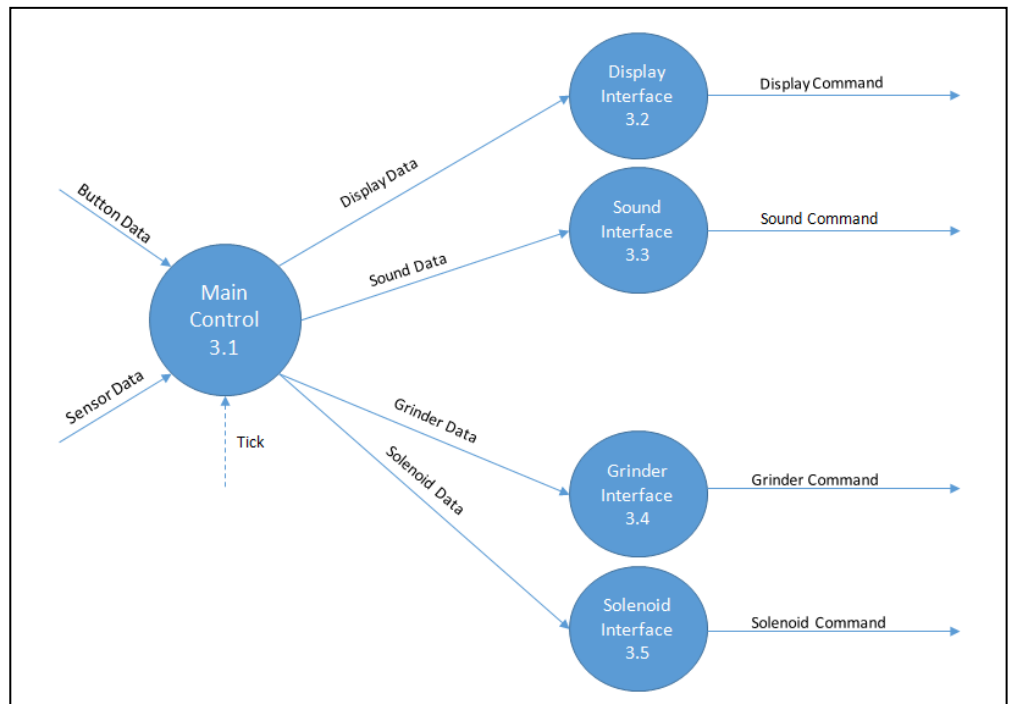
### 3.2.3 DFD Level 2

#### 3.2.3.1 DFD

##### 3.2.3.1.1 Button Receiver 1 & Sensor Receiver 2



3.2.3.1.2 Main Control 3



3.2.3.2 Process Specification

3.2.3.2.1 Process 1.1

<b>Reference No</b>	1.1
---------------------	-----

<b>Name</b>	Button Interface
<b>Input</b>	Extract Button Concentration Button Hot / Cold Button Supply Button Reservation (Clean) Button Reservation (Extract) Button Reservation Cancel Button Clean Button
<b>Output</b>	Button Raw Data
<b>Process Description</b>	1. 사용자가 입력하는 버튼 Interface로서 Raw Data를 Button Data Composer로 전달

## 3.2.3.2.2 Process 1.2

<b>Reference No</b>	1.2
<b>Name</b>	Button Data Composer
<b>Input</b>	Button Raw Data Tick
<b>Output</b>	Button Data
<b>Process Description</b>	1. Raw Data로 전달된 사용자 입력 값을 가공하여 Button Data형태로 전달

## 3.2.3.2.3 Process 2.1

<b>Reference No</b>	2.1
<b>Name</b>	Weight Sensor Interface
<b>Input</b>	Water Pool Weight Coffee Powder Weight
<b>Output</b>	Weight Data
<b>Process Description</b>	1. CM에서 사용되는 값 (물의 무게, 커피가루의 무게) 를 측정하여 Weight Data로 Sensor Data Composer에 전달

## 3.2.3.2.4 Process 2.2

<b>Reference No</b>	2.2
<b>Name</b>	Existence Sensor Interface
<b>Input</b>	Cup Existence
<b>Output</b>	Existence Data
<b>Process</b>	1. 커피를 담은 컵이 존재하는지 확인하여

<b>Description</b>	Existence Data로 Sensor Data Composer에 전달
--------------------	--

## 3.2.3.2.5 Process 2.3

<b>Reference No</b>	2.3
<b>Name</b>	Temperature Sensor Interface
<b>Input</b>	Water Pool (Hot) Temperature Water Pool (Cold) Temperature
<b>Output</b>	Temperature Data
<b>Process Description</b>	1. Cold Water Pool과 Hot Water Pool의 온도를 측정하여 Temperature Data를 전달

## 3.2.3.2.6 Process 2.4

<b>Reference No</b>	2.4
<b>Name</b>	Sensor Data Composer
<b>Input</b>	Weight Data Existence Data Temperature Data
<b>Output</b>	Sensor Data
<b>Process Description</b>	1. 각 센서로부터 전달받은 값을 struct 형태로 가공하여 전달

## 3.2.3.2.7 Process 3.1

<b>Reference No</b>	3.1
<b>Name</b>	Main Control
<b>Input</b>	Button Data Sensor Data
<b>Output</b>	Display Data Sound Data Grinder Data Solenoid Data
<b>Process Description</b>	<ol style="list-style-type: none"> <li>1. 사용자와 센서로부터 입력 받은 값을 가공한 데이터 구조체를 입력 받아 Output Interface에 전달</li> <li>2. Display Data로 화면에 표시할 데이터 구조체를 전달</li> <li>3. Sound Data로 소리로 전달할 데이터 구조체를 전달</li> <li>4. Grinder Data는 커피원두를 갈아야 하는</li> </ol>

	<p>지를 전달</p> <p>5. Solenoid Interface에 어떤 요소에 대한 양을 얼마나 조절하는지 구조체형태로 전달</p>
--	---

## 3.2.3.2.8 Process 3.2

<b>Reference No</b>	3.2
<b>Name</b>	Display Interface
<b>Input</b>	Display Data
<b>Output</b>	Display Command
<b>Process Description</b>	1. 사용자에게 feedback할 화면에 대한 정보를 처리

## 3.2.3.2.9 Process 3.3

<b>Reference No</b>	3.3
<b>Name</b>	Sound Interface
<b>Input</b>	Sound Data
<b>Output</b>	Sound Command
<b>Process Description</b>	1. 사용자에게 feedback할 소리에 대한 정보를 처리

## 3.2.3.2.10 Process 3.4

<b>Reference No</b>	3.4
<b>Name</b>	Grinder Interface
<b>Input</b>	Grinder Data
<b>Output</b>	Grinder Command
<b>Process Description</b>	1. 커피원두를 가는데 필요한 데이터를 처리

## 3.2.3.2.11 Process 3.5

<b>Reference No</b>	3.5
<b>Name</b>	Solenoid Interface
<b>Input</b>	Solenoid Data
<b>Output</b>	Solenoid Command
<b>Process Description</b>	<p>1. 구조체형태로 Solenoid Data를 전달받음</p> <p>2. 어떤 요소를 얼마나 추가하는지 정보를 처리해 정해진 양 만큼 CM에 들어갈 수 있도록 조절</p>

## 3.2.3.3 Data Dictionary

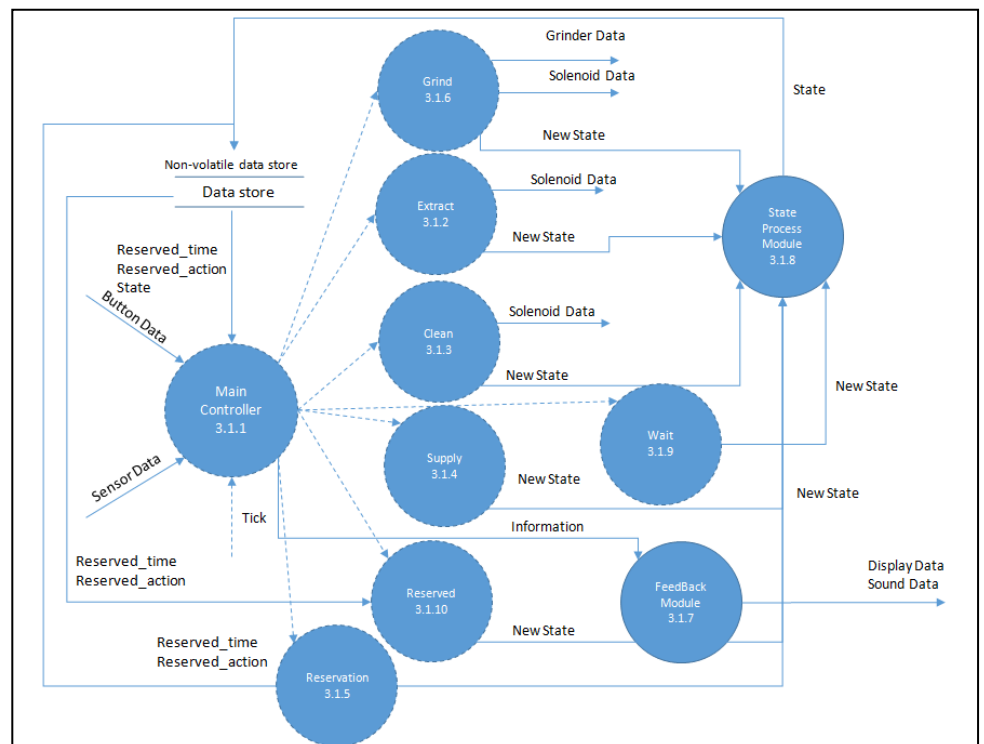


Input/ Output Event	Description	Format, Type
Button RawData	Push Button	Associative array [HIGH/ LOW], Periodic
Weight Data	Weight	Float, Periodic
Existence Data	Exist	Boolean, Periodic
Temperature Data	Temperature (Hot / Cold)	Hot / Cold, Periodic
Display Data	String for display	String, Periodic
Sound Data	Sound to play	On / Off, Periodic
Grinder Data	Data for grinder	On / Off, Periodic
Button Data[Clean]	Is clean button pressed	Boolean, Periodic
Solenoid Data	Data for solenoid	On / Off, Periodic

3.2.4 DFD Level 3

3.2.4.1 DFD

3.2.4.1.1 Main Control



3.2.4.2 Process Specification

3.2.4.2.1 Process 3.1.1

<b>Reference No</b>	3.1.1
---------------------	-------

<b>Name</b>	Main Controller
<b>Input</b>	Reserved Time Reserved Action State Button Data Sensor Data Tick
<b>Output</b>	None
<b>Process Description</b>	1. Tick 시간 단위로 Trigger되어 예약된 데이터가 존재하는지 확인하며 각 모듈을 Trigger 함

## 3.2.4.2.2 Process 3.1.2

<b>Reference No</b>	3.1.2
<b>Name</b>	Extract
<b>Input</b>	None
<b>Output</b>	Solenoid Data New State
<b>Process Description</b>	1. Main Controller로부터 trigger되어 커피를 추출(solenoid로 양을 조절) 2. Display로 사용자에게 feedback 3. 변경된 State를 적용함

## 3.2.4.2.3 Process 3.1.3

<b>Reference No</b>	3.1.3
<b>Name</b>	Clean
<b>Input</b>	None
<b>Output</b>	Solenoid Data New State
<b>Process Description</b>	1. Main Controller로부터 trigger되어 물을 Solenoid로 조절하여 CM을 청소 2. Display로 사용자에게 feedback 3. 갱신된 State를 적용,

## 3.2.4.2.4 Process 3.1.4

<b>Reference No</b>	3.1.4
<b>Name</b>	Supply
<b>Input</b>	None
<b>Output</b>	New State

<b>Process Description</b>	1. CM에 입력 받은 커피를 만들기 위해 필요한 요소를 공급
----------------------------	------------------------------------

## 3.2.4.2.5 Process 3.1.5

<b>Reference No</b>	3.1.5
<b>Name</b>	Reservation
<b>Input</b>	None
<b>Output</b>	Reserved Time Reserved Action New State
<b>Process Description</b>	1. 예약 요청이 들어오면 이 모듈을 통해 Data Store에 Reserved Time과 Reserved Action을 저장 2. 예약됨을 Display에 표시 3. 새롭게 State를 갱신

## 3.2.4.2.6 Process 3.1.6

<b>Reference No</b>	3.1.6
<b>Name</b>	Grind
<b>Input</b>	None
<b>Output</b>	Grinder Data Solenoid Data New State
<b>Process Description</b>	1. 원두커피 Grind 명령에 trigger되어 적절한 양의 원두를 Solenoid로 조절하여 갈도록 함 2. Display에 feedback 3. 새로운 state를 갱신

## 3.2.4.2.7 Process 3.1.7

<b>Reference No</b>	3.1.7
<b>Name</b>	Feedback Module
<b>Input</b>	Information
<b>Output</b>	Display Data Sound Data
<b>Process Description</b>	1. CM의 상태를 입력 받아 사용자가 이해할 수 있는 아날로그 정보로 변환

## 3.2.4.2.8 Process 3.1.8

<b>Reference No</b>	3.1.8
<b>Name</b>	State Process Module
<b>Input</b>	New State
<b>Output</b>	State
<b>Process Description</b>	1. 새로운 State를 갱신하도록 New State를 입력 받아 Data Storage에 State를 기록

3.2.4.2.9 Process 3.1.9

<b>Reference No</b>	3.1.9
<b>Name</b>	Wait
<b>Input</b>	None
<b>Output</b>	New State
<b>Process Description</b>	1. 새로운 State를 갱신하도록 New State를 입력 받아 Data Storage에 State를 기록 2. 대기중인 상태를 표시

3.2.4.2.10 Process 3.1.10

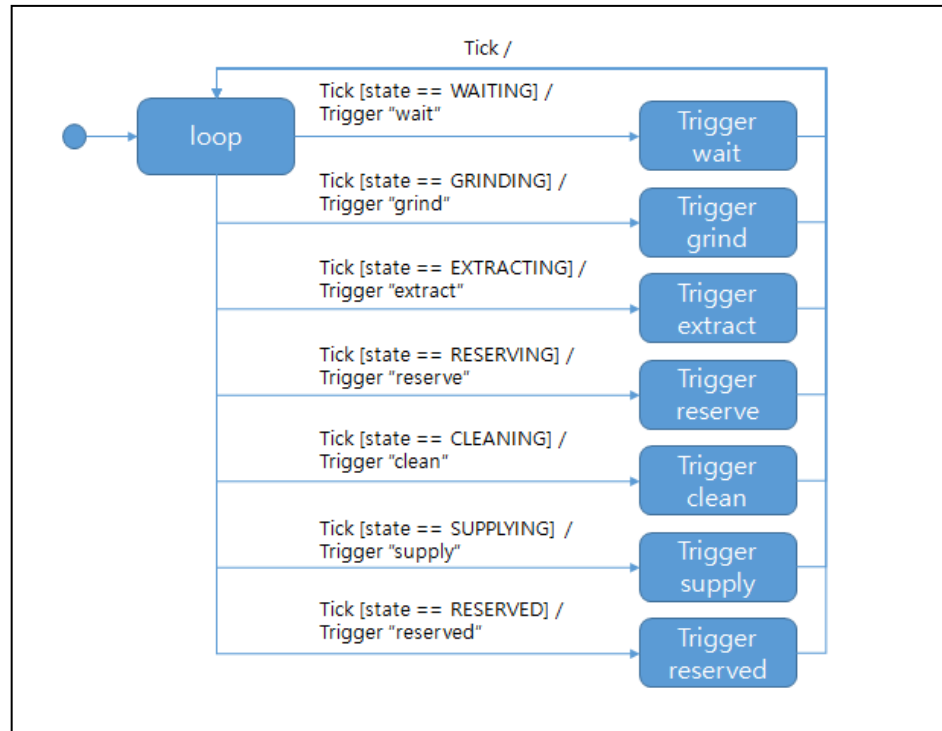
<b>Reference No</b>	3.1.10
<b>Name</b>	Reserved
<b>Input</b>	Reserved_time Reserved_action
<b>Output</b>	New State Reserved_time
<b>Process Description</b>	1. reserved_time과 reserved_action을 전달 받아 예약된 작업을 수행

3.2.4.3 Data Dictionary

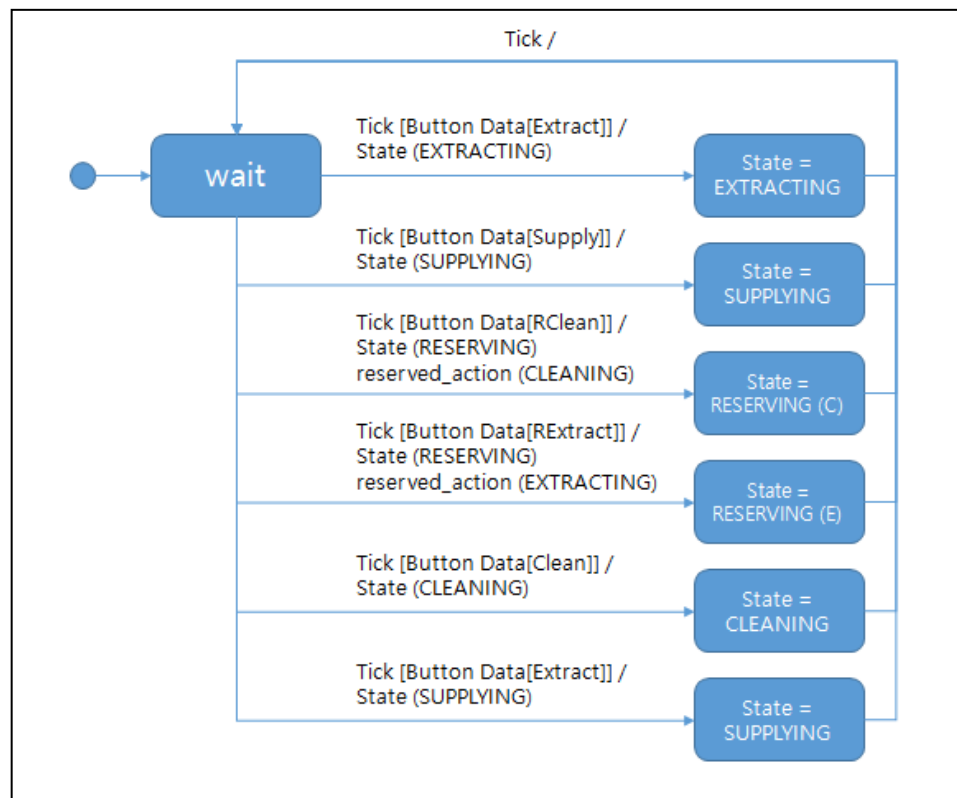
<b>Input/ Output Event</b>	<b>Description</b>	<b>Format, Type</b>
Reserved Time	사용자가 정한 예약 시간	struct date{ int year; int month; int day; int time; }
Reserved Action	사용자가 예약으로 요청한 동작	#define MK_COFFEE 0 #define CLEAN 1
State	현재 CM의 상태	#define IDLE 0

		<pre>#define WORKING 1 #define RESERVE 2 #define RESERVING 4</pre>
Grinder Data	원두를 가는데 필요한 데이터	<pre>struct grind_data{     float quantity;     bool is_exist; } //is_exist flag로 기존에 깔린 커피가루가 존재하는지 검사</pre>
Solenoid Data	커피, 물, 원두의 양을 조절하도록 하는 데이터	<pre>struct solenoid {     int element;     float quantity; }</pre>
Display Data	사용자에게 보여주기 위한 데이터	<pre>string disp; /* ex.) 물: 80% 원두 없음 [+]청소필요 */</pre>
Sound Data	사용자에게 소리로 표현하기 위한 데이터	<pre>sound n; //ex.) beep-beep</pre>
New State	새롭게 갱신될 현재 CM의 상태	Same as State
Information	CM에서 관여하는 모든 정보	<pre>struct info {     date reserved;     float cold_water;     float hot_water;     float coffee_bean;     int need_clean;     float concentration; }</pre>

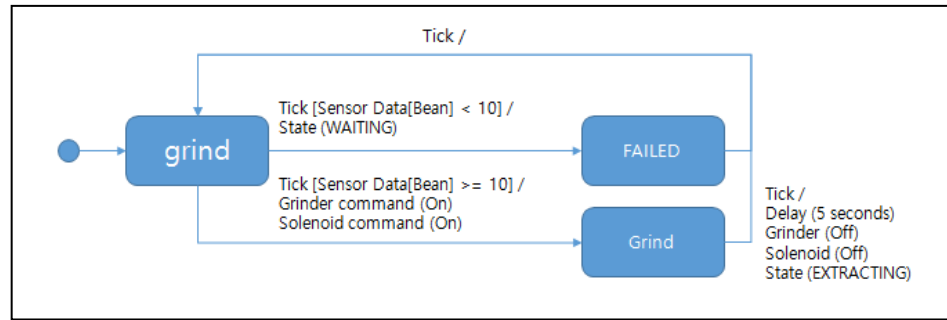
3.2.4.4 State Transition Diagram (*Main Controller*)



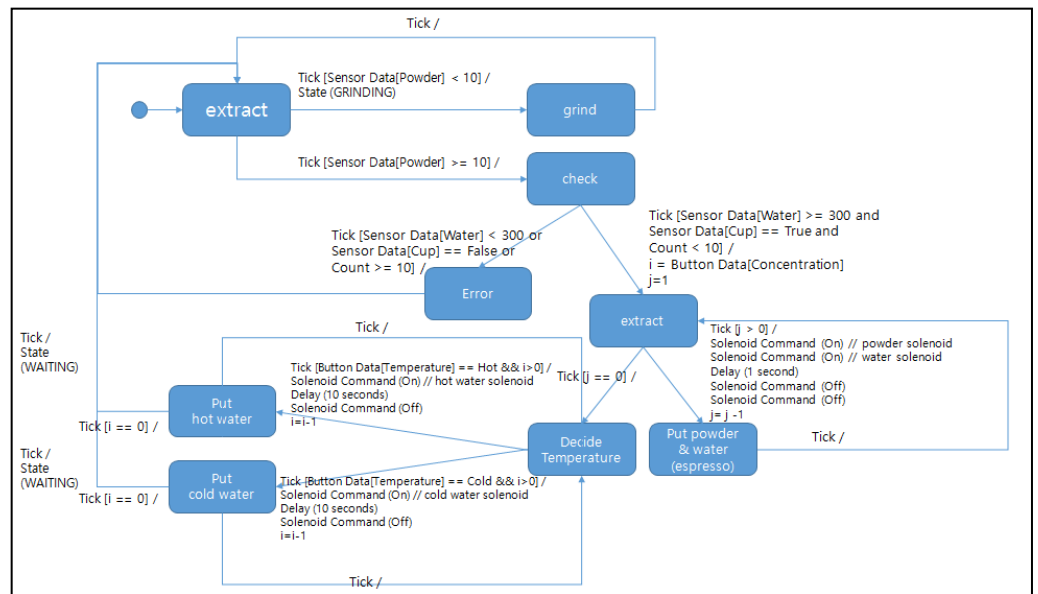
3.2.4.5 State Transition Diagram (*Wait*)



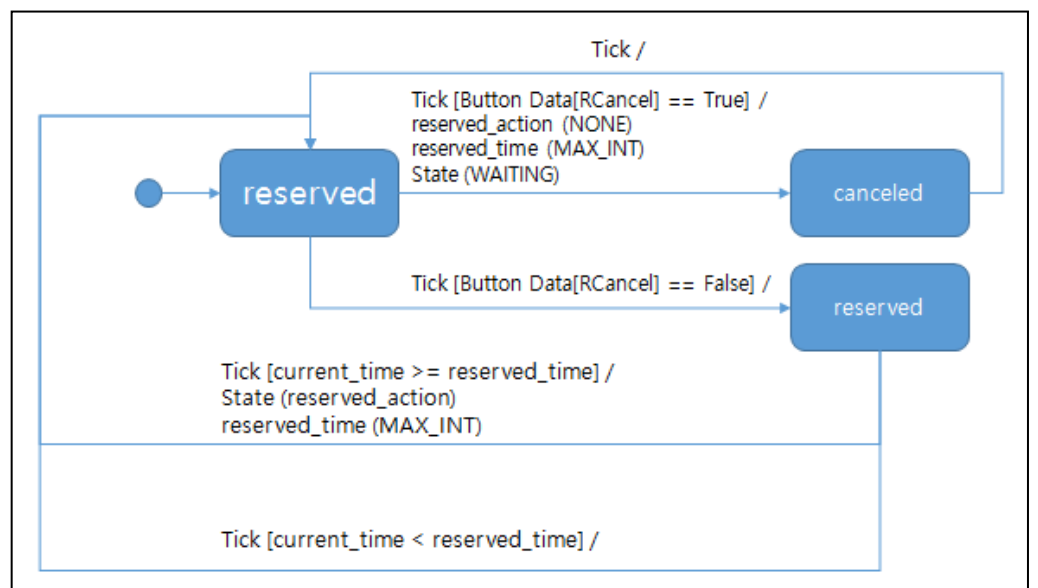
3.2.4.6 State Transition Diagram (*Grind*)



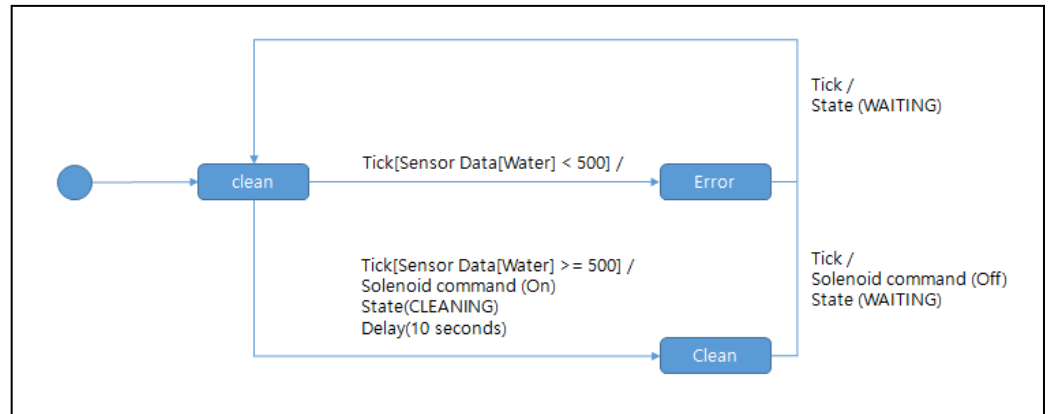
3.2.4.7 State Transition Diagram (*Extract*)



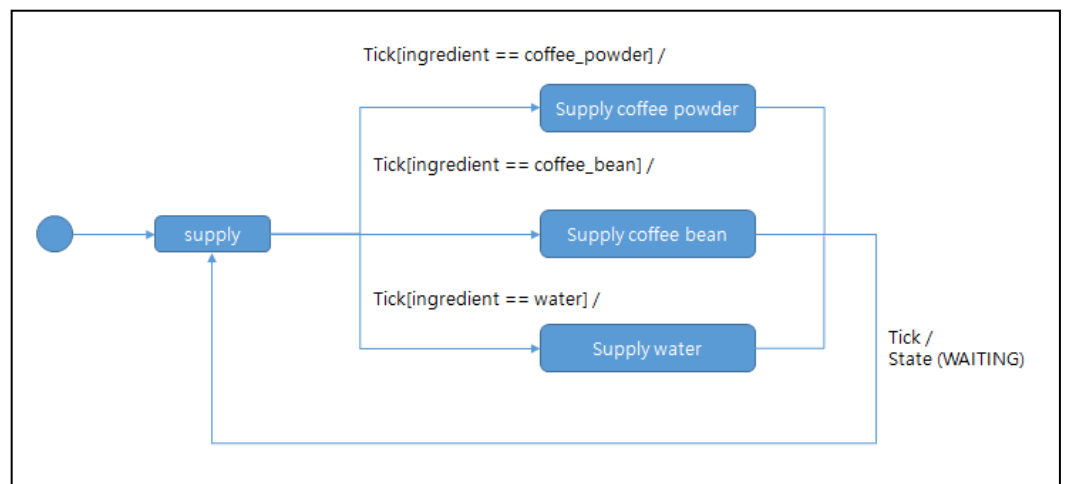
3.2.4.8 State Transition Diagram (*reserved*)



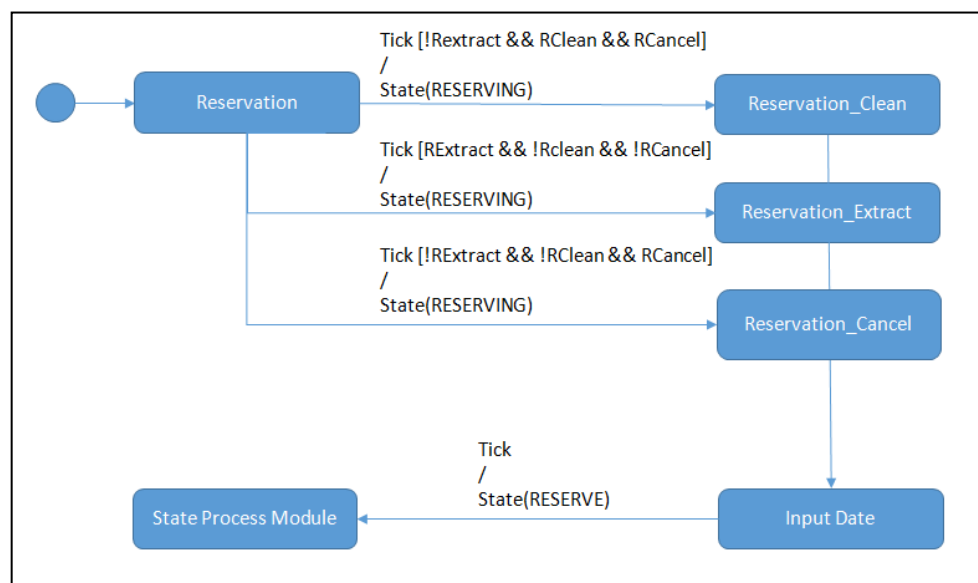
3.2.4.9 State Transition Diagram (*Clean*)



3.2.4.10 State Transition Diagram (*Supply*)



3.2.4.11 State Transition Diagram (*Reservation*)





### 3.2.5 Overall DFD

