

IEEE 830-1998 standards

SRS for the DVM controller

Practice #5 : SA/SRS for DVS

Team 3 성지운, 이동희, 이정, 정준원

2020.12.12

목차

1. Introduction

- 1.1 Purpose
- 1.2 Scope
- 1.3 Definitions, acronyms, and abbreviations
- 1.4 References
- 1.5 Overview

2. Overall description

- 2.1 Product perspective
- 2.2 Product functions
- 2.3 User Characteristics
- 2.4 Constraints
- 2.5 Assumptions and dependencies

3. Specific requirements (A.6)

- 3.1 External interface requirements
- 3.2 Functional requirements
- 3.3 Performance requirements
- 3.4 Design constraints
- 3.5 Software system attributes

1. Introduction

1.1 Purpose

이 문서는 DVM controller 의 개발에 대한 세부적인 분석 및 설명을 위해 작성되었다.

1.2 Scope

DVM controller 는 DVM 의 음성안내(Speaker), 정보 수신이나 전송(Network driver), 음료제공(provider), 안내문 출력(Printer), 결제 요청(Payment request)을 담당하는 SW 이다.

Code pad(HW), DVM Network driver(HW), Drink select button(HW), Card reader module(HW) 에게 info 를 받는 interface 부터 Guide message, DVM message output, Drink to provide, Content to print, Payment request 를 전달하는 interface 까지 DVM controller 로 지정한다.

(하단의 Overall description 의 그림 또는 SA 참고)

1.3 Definitions, acronyms, and abbreviations (대소문자 구분 없음)

- DVM: Distributed vending machine (SA 2페이지의 디자인 참고)
- DVM num: DVM의 고유 번호
- Msg, HW, SW, var: Message, Hardware, Software, Variable(Memory)의 약어
- Self, other, sth, F, T: 자기 자신(DVM), 다른 DVM, something (None과 구분하기 위함), False, True의 약어
- DVM controller: 이 프로젝트의 전반을 아우르는 SW, input interface부터 output interface까지 포함한다.
- Drink select button: 사용자가 음료를 선택하는 Button
- DVM Network driver: DVM간에 Network로 정보를 주고받기 위한 Driver. OSI의 1~6th layer에 해당하는 부분이지만(Application layer만 제외) 편의상 DVM Network

driver로 부르기로 한다. 입력과 출력을 구분하기 위해 Input DVM Network driver, Output DVM Network driver로 나눠서 용어를 사용하지만 서로 동일한 것이다.

- Code pad: 사용자 인증번호를 입력하는데 쓰이는 Pad.
- Card reader module: 카드를 통한 결제를 진행할 Module (COTS). 입력과 출력을 구분하기 위해 Input Card reader module, Output Card reader module로 나눠서 용어를 사용하지만 서로 동일한 것이다.
- Speaker module: 특정 Guide message를 받아서 사용자에게 안내하는 module. 자체적으로 재생할 음성 파일을 갖고 있음.
- Controller: var 및 list로부터 input들을 받아 종합적으로 계산하는 프로세스
- Drink provider: DVM controller로부터 정보를 받아서 사용자가 선택한 음료를 제공하는 HW
- Printer: DVM controller로부터 정보를 받아서 사용자에게 다른 DVM의 위치와 인증번호를 인쇄해주는 HW
- SpeCnt: DVM이 가지고 있는 '종류', '제공가능수량', '인증번호'를 저장하는 리스트
- Selected drink: Drink select button이 DVM Controller에게 전달하는 정보(bit string, 변환 시 음료 이름에 해당)
- DVM Message input: DVM Network driver가 DVM Controller에게 전달하는 Msg(String)
- Code input: Code pad가 DVM Controller에게 전달하는 정보(bit string, 변환 시 숫자 및 '*'에 해당)
- Payment verified: Card reader module이 DVM Controller에게 전달하는 리포트(bit string, 변환 시 결제 진행 결과 10 | 01 | 00에 해당)
- Guide message: DVM Controller가 Speaker module에게 전달하는 정보(000~111)
- DVM Message output: DVM Controller가 DVM Network driver에게 전달하는 Msg(String)
- Drink to provide: DVM Controller가 Drink provider에게 전달하는 정보(음료이름을 변환한 bit string)
- Content to print: DVM Controller가 Printer에게 전달하는 정보(프린트할 내용을 변환한 bit string)

- Payment request: DVM Controller가 Card reader module에게 전달하는 정보(결제 진행을 요청하는 bit string)
- Rank: 12자리로 이루어진 List로 다음과 같다. ([음료이름, DVM num1, DVM num2,..., DVM num9, 예약취소 message, 인증번호])
- Price list: 음료 20종의 이름과 가격을 적은 리스트.
- Near: 자신으로부터 가까운 순서대로 DVM num이 적혀 있는 리스트.
- Src_id: Source id의 약어, 전송인(DVM) num에 해당. 2글자 String이다.
Ex) DVM num = 7 → Src_id = 07
- Dst_id: Distant id의 약어, 수신인(DVM) num에 해당. Src_id와 동일한 형식.
- 'B', 'F', 'R', 'D': 'B'=broadcast, 'F'=feedback, 'R'=reserve, 'D'=delete
- Rank == N/A: 의미적으로 N/A이며, 실제 상태는 Rank == ['음료이름']를 의미한다.
- Rank 초기화: Rank==[]를 의미한다.

1.4 References

- IEEE 830-1998 (A.6)
- DVM SA (Class B Team 3)

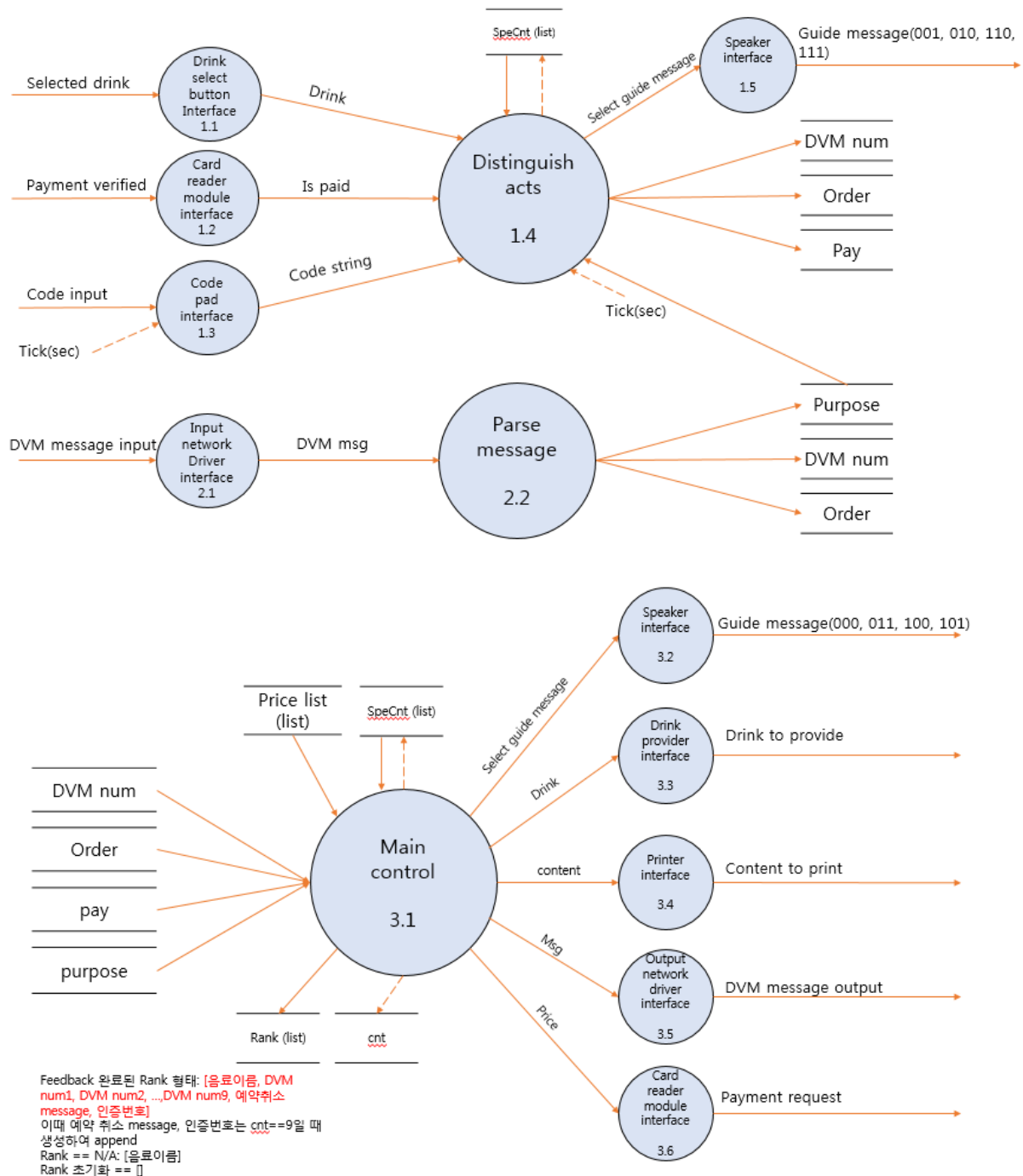
1.5 Overview

이 문서는 IEEE 830-1998 형식으로 작성되었다.

- Section 2: SW, SW의 기능, 사용자, 사용 및 개발 제한, 가정 및 종속성
- Section 3: 기능/비기능 요구사항.

2. Overall Description

이 SW는 DVM의 핵심적인 동작을 컨트롤하기 위해 디자인되었다. 사용자가 DVM을 조작하거나 다른 DVM으로부터 Msg를 받았을 때 DVM 내부의 정보와 입력받은 정보를 종합적으로 계산하는 프로그램이다.



2.1 Product perspective

이 SW 는 DVM 의 핵심적인 동작을 컨트롤하기 위해 디자인되었다. 사용자가 외부의 Code pad, Drink select button, card reader module 을 통해 조작하고, 다른 DVM 이 network 를 통해 Msg 를 전송하여 각각의 입력이 들어오면 그에 맞는 동작을 수행한다.

2.1.1 Hardware interfaces

Speaker module, Drink provider, Printer, Drink select button, Code pad 에 대한 interface 가 있다.

2.1.2 Software interfaces

COTS 인 card reader module 에 대한 interface 가 있다.

2.1.3 Communications interfaces

DVM 을 network 에 연결해주는 DVM Network driver 에 대한 interface 가 있다.

2.1.4 System interfaces

따로 관리하는 DB 가 없고, Fully connected network 형태로 hub 나 center 가 없으므로 System interface 는 없다. Stand-alone system.

2.1.5 User interfaces (사실상 없으나 의미적으로, HW, SW interfaces 에 포함)

Input interface 부터 Output interface 까지만 다루므로 DVM Controller 자체에는 End user interaction 이 없으나, End user 가 input 을 제공한다는 관점에서 Drink select button interface, Code pad interface, Input Card reader module interface, 즉 end user 가 조작하는 HW 에 대한 interface 가 있다.

2.2 Product functions

이 영역은 DVM controller 의 기능에 대해 서술한다.

2.2.1 Drink select button 으로부터 버튼 입력 인식

Selected drink 를 interface 를 통해 input 으로 받고, 해당하는 음료의 이름(string)으로 변환한다. 음료 이름을 Order (var)에, 자신의 DVM num 을 DVM num (var)에 저장하고, Pay (var)는 F(False)로 바꾼다. 이때, SpeCnt 에서 해당 음료의 제공가능수량을 -1 한다.

2.2.2 다른 DVM 으로부터 msg 수신

다른 DVM 이 전송한 Msg 를 Input DVM Network driver interface 를 통해 input(string)으로 받는다. 받은 Msg 를 Src_id + dst_id + '음료이름'(String) + 'Purpose' ('B'|'F'|'R'|'D'|('*로 시작하는 6 글자 string))로 parsing 하여 Src_id 는 DVM num (var)에, '음료이름'은 Order (var)에, 'Purpose'는 Purpose (var)에 저장한다. Dst_id 는 사용하지 않는다.

2.2.3 Card reader module 로부터 카드 결제 정보 인식

Card reader module 이 결제 여부에 따라 다음과 같은 output 을 주고, DVM Controller 는 다음과 같은 동작을 수행한다.

결제성공 → 10: Pay 를 T(True)로 바꾼다.

시간초과 → 01, 잔고부족 → 00: 시간 초과(001) 및 잔고 부족(010)에 해당하는 bit string 으로 변환하여 Speaker 에 전달한다. SpeCnt 의 제공가능수량을 +1 한다.
(Pay 는 사용자가 버튼을 눌렀을 때 이미 F 상태이고, 바뀌는 경우는 없다.)

2.2.4 Code pad 로부터 Code input 인식

Code pad 를 통해 사용자로부터 Code input 을 받을 때마다 interface 에서 순서대로 변환한다. 이때, Code input 의 변환형이 '*'일 경우, '*'을 포함한 연속된 6 번의 입력을 하나의 String 으로 연결한다. 만약 30 초 내에 6 자리를 완성하지 못할 경우(누르다 만 것으로 간주) *00000 으로 변환한다.

완성된 string 을 SpeCnt 에서 탐색하여 일치하면 해당하는 음료이름을 Order (var)에 저장하고, DVM num (var)에는 자신의 DVM num 을 저장한다. 또한 Pay 는 T 로 바꾼다.

완성된 String 이 SpeCnt 에 없다면 인증번호 오류(110)에 해당하는 bit string 으로 변환하여 Speaker 에 전달하고 모든 프로세스를 초기화한다.

2.2.5 Guide message 생성

Speaker interface 로 전달된 내용(000~111)을 Speaker module 에게 전달

2.2.6 Drink provider 에게 제공할 Drink to provide 생성

Code pad 나 Drink select button 에 의해 전달받은 음료수를 자신(DVM)이 제공해야할 경우 비용 지불이 확인되면 Drink to provide 를 생성하여 Drink provider 에게 전달한다.

2.2.7 Printer 에게 전달할 Content to print 생성

Drink select button 으로 선택된 음료수를 계산했으나 내가(DVM) 제공할 수 없을 경우 대신 제공하기로 한 DVM num 과 인증번호를 포함한 Content to print 를 생성하여 Printer 에게 전달한다.

→ DVM num + "번 자판기에서 가져가세요.\n 인증번호: " + 인증번호

2.2.8 DVM Network driver 에게 전달할 DVM message output 생성

상황에 따라 Broadcast | Feedback | Delete | Reserve | 선결제 Msg 를 생성하여(DVM message output) Output DVM Network driver 에게 전달한다.

2.2.9 Card reader module 에게 전달할 Payment request 생성

결제가 필요할 경우 Payment request 를 Output Card reader module 에게 전달한다.

2.3 User Characteristics

이 SW 에는 2 종류의 user 가 있다: 일반 사용자, 개발자.

→ 일반 사용자는 DVM controller 의 존재를 모르며, DVM 과 연결된 HW (Drink select button, Code Pad, Input Card reader module)를 통해 DVM controller 의 Input 을 생성할 수 있다.

→ 개발자는 DVM controller 의 SW 에 수정 및 업그레이드가 발생할 경우 DVM controller 의 코드를 직접 조작할 수 있다.

2.4 Constraints

이 SW 는 다음의 제약 조건을 준수하여 코딩해야 한다.

- 이 SW 는 Python 언어로 코딩한다.
- 모든 tick 은 sec(초)로 가정한다.
- 일반 사용자가 DVM controller 에 직접적으로 접근 및 조작할 수 없다.

2.5 Assumptions and dependencies

- DVM controller 는 아두이노 기반으로 작동한다.
- Card reader module 은 COTS 로, 사용자가 카드를 입력하는 순간부터 VAN 사업자와 통신 및 결과 리포트(결제완료 | 잔고부족)까지를 담당한다. 또한 Payment request 를 받고나서 20 초를 스스로 세서, 20 초안에 결제가 되지 않으면 시간초과를 반환한다고 가정한다. Card reader module 이 DVM controller 에게 전달하는 output 은 결제완료, 잔고부족, 시간초과 밖에 없다.
- Speaker module 에는 8 종류의 안내가 녹음되어 있어서 단순히 각 음성에 대한 Bit string 을 trigger 로 받으면 각 음성을 재생한다.
- DVM Network 에게 전달하는 정보는 DVM controller(OSI 7th layer)가 캡슐화 등을 하지 않고 DVM Network driver(OSI 1~6th layer)에서 자동적으로 진행하도록 한다.
- SpeCnt (list)에 저장되어 있는 인증번호를 추가로 저장할 때 중복되는 값이 존재하면 중복 방지를 위해 덮어쓴다.
- 선결제 안내 시 22 초 이내에 결제가 이루어지지 않으면 고객이 떠난 것이라고 간주하고 시간초과 및 예약 취소를 진행한다.
- Broadcast Msg 를 받으면 Drink select button 에 대해 5tick 간 대기한다. Drink select button interface 와 var 사이의 process 가 대기를 수행한다. (Broadcast Msg 수신 → Feedback Msg 전송 → 예약요청 Msg 수신 → 제공가능수량 -1 의 과정이 5tick 안에 이루어지며, 그동안 사용자가 drink select button 을 조작해 제공가능수량에 충돌이 생기지 않도록 하기 위함이다.)
- 각 과정에서 "사람이 n 초 이내에 수행하지 못할 경우"의 의미는 행동학적으로 99%의 사람에게 n 초가 해당 행위를 하기에 충분한 시간이라는 가정 하에 쓰여진 것이다. 또한 DVM 이 broadcast 를 전송할 때부터 예약을 마무리 할 때까지(선결제를 의미하는 것이 아닌 단순히 제공가능수량 -1 하는 것을 말함) 5 초면 충분하다고 가정한다.

- Network msg 와 Guide msg 의 종류 및 형태는 다음과 같다.

Msg name	Structure
Broadcast	Src_id + 00 + 음료이름 + 'B' ex)0700사이다B (나 07인데 사이다 있는 DVM?)
Feedback	Src_id + dst_id + (음료이름 'None') + 'F' ex)0107사이다F (나 01인데 사이다 있어 07아) /// 0507NoneF (나 05인데 사이다는 없어 07아)
예약 요청	Src_id + dst_id + 음료이름 + 'R' ex)0701사이다R (01아 나 07인데 사이다 하나만 예약해줘 = 사이다 제공 가능 수량 -1해줘)
예약 취소	Src_id + dst_id + 음료이름 + 'D' Ex)0701사이다D (01아 나 07인데 사이다 예약한 거 취소해줘 = 사이다 제공 가능 수량 +1해줘)
선결제	Src_id + dst_id + 음료이름 + 인증번호 ex)0701사이다*25001 (01아 나 07인데 사이다 인증번호는 *25001이니까 사이다 인증번호 list에 추가해 줘)

Guide msg	meaning	Guide msg	Meaning
000	카드 결제: "결제할 카드를 리더기에 입력해주세요 "	100	선결제: "현재 자판기에 없는 음료입니다. 다른 자판기에서 가져가실 수 있도록 선결제를 도와드리겠습니다. 결제할 카드를 리더기에 입력해주세요."
001	시간 초과: "시간이 초과되었습니다. 다시 선택해주세요."	101	전체 재고 소진: "해당 음료는 모두 소진되었습니다. 다시 선택해주세요."
010	잔고 부족: "계좌의 잔액이 부족합니다."	110	인증번호 오류: "인증번호 오류입니다. 다시 입력해주세요."
011	음료 제공 완료: "음료 제공이 완료되었습니다."	111	"잠시 기다려주세요"→ Broadcast를 받았을 때 부터 예약 요청 msg를 받을 때 까지는 해당 음료 button이 눌리면 안됨. 5초면 충분하다고 가정.

3. Specific requirements

3.1 External interface requirements

3.1.1 User interfaces

3.1.1.1 Drink select button interface

User가 입력한 Drink select button의 결과 값을 input으로 받는다.

input을 interrupt하게 받고 bit string 타입의 Input을 "Drink"라는 string 타입의 output으로 변환한 후 "Distinguish acts"에게 전달한다.

3.1.1.2 Code pad interface

tick과 User가 입력한 Code pad의 결과 값을 input으로 받는다.

input을 interrupt하게 받고 bit string list 타입의 input을 "Code String"라는 string 타입의 output으로 변환한 후 "Distinguish acts"에게 전달한다.

만약, 30초내에 수행하지 못하면 '*00000' 값인 "Code string"을 output으로 변환한다.

3.1.1.3 Input card reader module interface

User가 결제한 후 결제에 대한 card reader module의 결과 값을 input으로 받는

다.

input을 interrupt하게 받고 bit string list 타입의 input을 "Is paid"라는 string 타입의 output으로 변환한 후 "Distinguish acts"에게 전달한다.

3.1.1.4 Distinguish acts (Process)

Drink, Is paid, Code string, SpeCnt (list), purpose (var), tick을 input으로 받는다.

각 input에 대해 다음 과정을 진행한다.

Drink, SpeCnt (list), Purpose (var)에 대해

요청받은 음료의 이름(Drink)을 SpeCnt의 '종류'에서 탐색하고

1. 나에게 없으면 메모리 DVM num (var)에 0 저장
2. 나에게 있지만 해당 음료의 "제공 가능 수량"을 확인해보니 0개면 메모리 DVM num (var)에 0 저장
3. 나에게 종류가 있으며 재고도 있으면 SpeCnt (list)의 "제공가능수량"을 -1하고, 메모리 DVM num (var)에 자신의 DVM 번호 저장, Pay는 F로 만들
4. Drink는 위 1, 2, 3에 대해 모두 메모리 Order (var)에 동일하게 전송
5. Purpose의 정보가 'B'가 되면 위 1, 2, 3, 4번에 대해 바로 5초간 대기

Is paid에 대해

Is paid의 값이 01이면 "시간초과 guide message", 00이면 "잔고부족 guide message"에 해당하는 정보, 즉 000~111타입의 "Select guide message" (data)를 Speaker interface로 전달한다.

Code string, SpeCnt에 대해

Code string의 값을 SpeCnt의 '인증번호'와 비교하여 찾는다.

1. 동일한 코드가 존재하지 않으면 "110"(코드오류)를 000~111타입의 "Select guide message"로 Speaker interface로 전달 및 Code pad로부터 진행된 모든 프로세스를 초기화
2. 코드가 존재할 경우 해당 코드가 위치한 음료 이름(SpeCnt '종류')을 Drink

(data)로 전송 및 Order (var)에 저장, DVM num (var)에 자신의 DVM 번호 저장, Pay (var)는 T로 만듦

3.1.2 Hardware interfaces

3.1.2.1 Output card reader module interface

Main control이 전달하는 Price를 input으로 받는다.

input을 interrupt하게 받고 int 타입의 input을 "Payment request"라는 Bit string 타입의 output으로 변환한 후 "Card reader module"에게 전달한다.

3.1.3 Software interfaces

3.1.3.1 Speaker interface

Main control 또는 Distinguish act가 전달하는 Select guide message를 input으로 받는다.

input을 interrupt하게 받고 000~111 타입의 input을 "Guide message"라는 000~111 타입의 output으로 "Speaker module"에게 전달한다.

3.1.3.2 Drink provider interface

Main control이 전달하는 Drink를 input으로 받는다.

input을 interrupt하게 받고 string 타입의 input을 "Drink to provide"라는 Bit string으로 변환한 후 "Drink provider"에게 전달한다.

3.1.3.3 Printer interface

Main control이 전달하는 content를 input으로 받는다.

input을 interrupt하게 받고 string 타입의 input을 "Content to print"라는 Bit string 타입의 output으로 변환한 후 "Printer"에게 전달한다.

3.1.4 Communications interfaces

3.1.4.1 Input DVM Network Driver interface

다른 DVM로부터 MSG를 받은 DVM Network driver의 결과 값을 input으로 받는다.

input을 interrupt하게 받고 string 타입의 input을 string 타입의 "DVM message"로 "Parse message"에게 전달한다.

3.1.4.2 Output network Driver interface

“Main control”이 전달한 Msg를 input 으로 받는다.

input을 interrupt하게 받고 string 타입의 input을 string 타입의 “DVM message output”로 “DVM Network Driver”에게 전달한다.

3.1.4.3 Parse message (Process)

“Input network Driver interface”가 전달한 DVM msg를 input으로 받는다.

input을 interrupt하게 받고 string 타입의 input을 parsing하여 각각 메모리 (var)에 저장한다.

앞 2자리(Src_id) : DVM num (var)에 저장

그 뒤 2자리(dst_id) : 버림

음료이름 : Order (var)에 저장

‘B’ | ‘F’ | ‘R’ | ‘D’ | (*로 시작하는 문자열) : purpose (var)에 저장

3.2 Functional requirements – Controller가 받는 stimulus 기준.

3.2.1 Stimulus 1 (situation: charge for my drink)

No.	Name	Value
1	DVM num	self
2	Order	sth
3	pay	F

3.2.1.1 Functional requirement 1.1

사용자가 Drink select button을 통해 원하는 음료를 고른다.

이후 Drink select button interface가 Distinguish acts로 Drink (data)를 전달한다.

3.2.1.2 Functional requirement 1.2

Distinguish acts가 Drink를 받은 후 해당 음료가 자신에게 있는지 SpeCnt로 확인한다.

만약 음료의 종류가 자신에게 있고 재고도 존재한다면 메모리 DVM num (var)에 자신의 DVM num(self)을 저장하고 메모리 pay (var)에는 아직 결제하지 않았다는 F를 저장한다.

3.2.1.3 Functional requirement 1.3

위 var들의 stimulus에 의해 Controller는 전달받은 Drink를 메모리 Price list에서 탐색하여 해당 음료의 가격을 도출하고 Card reader module interface로 전달한다.

이후, Speaker interface로 카드 결제 음성 안내(000)을 전송한다.

3.2.2 Stimulus 2 (situation: My drink is paid)

No.	Name	Value
1	DVM num	self
2	Order	sth
3	pay	T

3.2.2.1 Functional requirement 2.1

앞서 사용자가 Drink select button을 통해 원하는 음료를 고르고 해당 정보를 Distinguish acts가 받아 메모리 Order (var)에 저장했으며 사용자가 고른 음료가 본인에게 종류도 존재하고 재고도 존재하여 DVM num이 self num인 상태이다.

3.2.2.2 Functional requirement 2.2

사용자가 결제를 하고 Card reader module interface가 전달한 값이 성공(10)일 때 Distinguish acts는 해당 정보를 받은 후 메모리 pay에 T를 저장한다.

3.2.2.3 Functional requirement 2.3

위 var들의 stimulus에 의해 Controller는 전달받았던 Drink를 그대로 Drink Provider interface로 전송한다.

이후, Speaker interface로 음료 제공 완료 음성 안내(011)을 전송한다.

3.2.3 Stimulus 3 (situation: Send Broadcast – I don't have one)

No.	Name	Value
1	DVM num	0
2	Order	sth
3	pay	F

3.2.3.1 Functional requirement 3.1

사용자가 Drink select button을 통해 원하는 음료를 고른다.

이후 Drink select button interface가 Distinguish acts로 Drink (data)를 전달한다.

3.2.3.2 Functional requirement 3.2

Distinguish acts가 Drink를 받은 후 해당 음료가 자신에게 있는지 SpeCnt로 확인한다.

음료의 종류가 자신에게 없어서 메모리 DVM num (var)에 0을 저장하고 메모리 pay (var)에는 아직 결제하지 않았다는 F를 저장한다.

3.2.3.3 Functional requirement 3.3

위 var들의 stimulus에 의해 Controller에서는 전달받은 Drink를 이용해 My number + 00 + 음료 이름 + 'B' 형태의 broadcast msg를 만들어 Output network driver interface로 전송한다.

이후, cnt를 0으로 만들고 Rank에 음료 이름을 append한다. = (Rank ==N/A, 1.3 참고)

3.2.4 Stimulus 4 (situation: send feedback)

No.	Name	Value
1	DVM num	other
2	Order	sth
3	pay	F
4	Purpose	'B'

3.2.4.1 Functional requirement 4.1

Input network driver interface가 DVM Network driver로부터 DVM Message input을 받아 DVM msg로 변환하여 Parse message로 전달한다..

3.2.4.2 Functional requirement 4.2

Parse message는 Input network driver interface로부터 받은 DVM msg를 parsing하여 각 var에 저장한다

이때, DVM num에는 broadcast를 보낸 DVM number가 저장되고

Order에는 음료 이름, pay에는 F, Purpose에는 B가 저장된 경우이다.

3.2.4.3 Functional requirement 4.3

위 var들의 stimulus에 대해 Controller는 전달받은 Drink를 SpeCnt에서 탐색하여 있으면 음료 이름, 없으면 'None'으로 변환하여 My number + DVM num + (음료 이름 | 'None') + 'F'형태로 feedback message를 만들어 Output network driver interface로 전송한다.

3.2.5 Stimulus 5 (situation: Other DVM wants to reserve mine)

No.	Name	Value
1	DVM num	other
2	Order	sth
3	Purpose	'R'

3.2.5.1 Functional requirement 5.1

Input network driver interface가 DVM Network driver로부터 DVM Message input 을 받아 DVM msg로 변환하여 Parse message로 전달한다..

3.2.5.2 Functional requirement 5.2

Parse message는 Input network driver interface로부터 받은 DVM msg를 parsing 하여 각 var에 저장한다

이때, DVM num에는 "예약요청 msg"를 보낸 DVM number (Src_id)가 저장되고

Order에는 음료 이름, Purpose에는 R가 저장된 경우이다

3.2.5.3 Functional requirement 5.3

위 var들의 stimulus에 의해 Controller는 전달받은 Drink를 SpeCnt에서 탐색하여 해당 음료의 '제공가능수량'을 -1한다.

3.2.6 Stimulus 6 (situation: Other DVM wants to delete its reservation)

No.	Name	Value
1	DVM num	other
2	Order	sth
3	Purpose	'D'

3.2.6.1 Functional requirement 6.1

Input network driver interface가 DVM Network driver로부터 DVM Message input 을 받아 DVM msg로 변환하여 Parse message로 전달한다..

3.2.6.2 Functional requirement 6.2

Parse message는 Input network driver interface로부터 받은 DVM msg를 parsing 하여 각 var에 저장한다

이때, DVM num에는 "예약취소 msg"를 보낸 DVM number가 저장되고

Order에는 음료 이름, Purpose에는 D가 저장된 경우이다

3.2.6.3 Functional requirement 6.3

위 var들이 stimulus에 대해 Controller에서는 전달받은 Drink를 SpeCnt에서 탐색하여 해당 음료의 '제공가능수량'을 +1한다.

3.2.7 Stimulus 7 (situation: 인증번호 for the reservation is arrived)

No.	Name	Value
1	Order	sth
2	Purpose	*____ (*로 시작하는 6자리 string)

3.2.7.1 Functional requirement 7.1

Input network driver interface가 DVM Network driver로부터 DVM Message input을 받아 DVM msg로 변환하여 Parse message로 전달한다.

3.2.7.2 Functional requirement 7.2

Parse message는 Input network driver interface로부터 받은 DVM msg를 parsing하여 각 var에 저장한다

이때, Purpose (var)에는 "선결제 msg"에 포함된 '*로 시작하는 6자리 string (ex.*12345)이 저장되고, Order (var)에는 음료 이름이 저장된 경우이다

3.2.7.3 Functional requirement 7.3

위 자극에 대해 Controller에서는 purpose (var)에 해당하는 인증번호를 자신 (DVM)의 SpeCnt의 '해당 음료이름'의 '인증번호'에 추가한다.

이 때 중복되는 값이 있으면 덮어쓴다.

3.2.8 Stimulus 8 (I am receiving feedbacks from other DVMs)

No.	Name	Value
1	DVM num	other
2	pay	F
3	Purpose	'F'

3.2.8.1 Functional requirement 8.1

Input network driver interface가 다른 DVM로부터 Feedback message를 받고 이를 parsing하여 각 var에 저장한다.

이 때, DVM num (var)에는 Feedback을 보낸 DVM num이 저장되고

Order에는 음료 이름, pay에는 F, Purpose에는 'F'가 저장되어야 한다.

3.2.8.2 Functional requirement 8.2

이 stimulus에 대해 Controller는 해당 메시지를 보고 Rank 리스트에 결과를 저장한다.

Rank input의 Drink가 'None'이면 해당 msg 무시, Drink가 음료이름이면 rank에 해당 feedback을 보낸 DVM num (Src_id)을 추가한다.

추가할 때 마다 Near (list)을 통해 Rank[1:]를 정렬한다.

이때, cnt가 (존재하는 모든 DVM의 수) -1 이 아니라면 Controller는 Rank 리스트를 추가하는 것을 반복한다.

3.2.9 Stimulus 8.1 (All feedbacks are received and turned out to have no supplies)

No.	Name	Value
1	Cnt	(존재하는 모든 DVM의 수) -1
2	Rank	N/A

3.2.9.1 Functional requirement 8.1.1

main controller가 Rank 리스트를 모두 완성한 경우이다.

Cnt == (존재하는 모든 DVM의 수) -1 이 확인되고 1 tick 이후 Rank (list) == N/A면 trigger를 받아 Select guide message에 전체 재고 소진(101)을 전송한다.

3.2.10 Stimulus 8.2 (All feedbacks are received and turned out to have supplies)

No.	Name	Value
1	pay	F
2	Cnt	(존재하는 모든 DVM의 수) -1
3	Rank	! N/A

3.2.10.1 Functional requirement 8.2.1

main controller가 Rank 리스트를 모두 완성한 경우이다.

Cnt == (존재하는 모든 DVM의 수) -1 이 확인되고 1 tick 이후 rank (list) != N/A면 Rank[0](음료이름)을 Price list와 비교하여 Price를 Card reader module로 전송한다. Speaker interface로 선 결제 안내 음성(100)을 전송한다.

또한, "예약취소 msg"를 생성하여 Rank[-2]에, 인증번호를 생성하여 Rank[-1]에 저장한다.

3.2.10.2 Functional requirement 8.2.2

이후 main controller에서 Cnt == (존재하는 모든 DVM의 수) -1, pay == F인 상

태로 22초가 지나면 Controller가 Rank[-2] (예약취소 msg)를 넘겨서 예약취소 msg를 Output Network driver interface에 전송한다.

이후, 모두 마무리 되면 cnt와 rank (list)를 초기화 한다.

3.2.11 Stimulus 8.3 (Prepay is done)

No.	Name	Value
2	Rank	!N/A
3	Cnt	(존재하는 모든 DVM의 수) -1
4	Pay	T

3.2.11.1 Functional requirement 8.3.1

고객이 선결제를 완료한 경우이다.

위 stimulus에 대해 Controller는 Printer에 전송할 Content to print를 생성하여 printer에 전달한다. 또한 "선결제 msg"를 만들어 Output network driver interface에 전송한다. 모두 마무리되면 cnt와 Rank (list)를 초기화한다.

Content to print = DVM num + "번 자판기에서 가져가세요.\n인증번호: " + 인증번호

3.3 Performance requirements

3.3.1 Performance

Tick에 따라 동작을 멈추고 실행하는 일이 있으므로 최대 5초 내에만 명령을 수행할 수 있으면 된다.

3.4 Design constraints

- 개발자는 자판기에 사용될 Embedded 장비에 올릴 Python으로 코딩해야 한다.

3.5 Software system attributes

3.5.1 Safety/Security

사용자 정보가 포함되지 않고 단지 결제 대행사로부터 받은 결제 성공 여부를 사용하므로 safety 문제가 없다.

3.5.2 Modularity

여러 모듈을 사용해서 하나의 시스템을 만들어서 modularity가 높다.

3.5.3 Scalability / Maintainability

어느 한 프로세스가 모든 일을 처리하는 것이 아니라 여러 프로세스가 각자 일을 하여 처리하므로 확장성 및 유지보수성이 좋다