|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| **사업명** | **정보통신방송연구개발사업** | | | |
| **과제명** | **상대방의 감성을 추론, 판단하여 그에 맞추어 대화하고 대응할 수 있는 감성지능 기술 연구개발** | | | |
| **문서명** | **소프트웨어 설계서** | | | |
|  | | | | |
|  | | **버전** | | **1.6** |
|  | | **일자** | | **2019년 01월 09일** |
|  | | **공개 범위** | **내부 공유** | |
| C:\Users\jinhong\Desktop\KAIST_1.gif | | | | |
|  | | | | |
| 본 문서에 대한 모든 저작권은 한국과학기술원에게 있습니다. 본 문서를 복사, 배포하거나 혹은 본 내용을 한국과학기술원의 서명 동의 없이 제3자에게 발설 및 배포하는 것은 금지되어 있습니다.  위의 위반으로 발생하는 모든 직.간접적 피해는 해당 위반자에게 있습니다. | | | | |

**문서 서명**

본 문서에 대한 서명은 한국과학기술원 내부에서 본 문서에 대하여 작성 및 유지 관리의 책임이 있음을 인정함

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 구분 | 소속 | 성명/직책 | 서명 | 일자 |
| 작성자 | 참여 기관 | 김준우 / 연구원 |  |  |
| 참여 기관 | 정선진 / 박사과정 |  |  |
| 참여 기관 | 정성우 / 석사과정 |  |  |
| 참여 기관 | 제갈수민 / 석사과정 |  |  |
| 검토자 | 해당 기관 책임자 | 노준용 / 교수 |  |  |
| 승인자 | 총괄 책임자 | 이수영 / 교수 |  |  |

**문서 이력**

|  |  |  |  |
| --- | --- | --- | --- |
| 버전 | 제/개정 일자 | 제/개정 위치 및 내용 | 작성자 |
| 0.5 | 2018/08/08 | 초안 작성 | 김준우 외 |
| 1.0 | 2018/08/14 | 2차년도 기본 설계서 배포 | 4세부 총괄 |
| 1.1 | 2018/08/20 | 3차년도 기본 설계서 작성 (감정기반 표정/입 생성 부문) | 정선진 외 |
| 1.2 | 2018/08/22 | RESTful API 및 입출력 규약 관련 요청사항 반영 | 정선진 외 |
| 1.3 | 2018/11/08 | 입출력 규약 명세 상세화 | 정선진 외 |
| 1.4 | 2019/01/02 | 모바일 부분 입출력 및 호출 절차 상세화 | 정선진 외 |
| 1.5 | 2019/01/03 | 서버 부분 입출력 및 호출 절차 상세화 | 정선진 외 |
| 1.6 | 2019/01/09 | 모바일 부분 호출 절차 수정 및 상세화 | 정선진 외 |

**목 차**

1. SW 설계 사항 5

1.1. 소프트웨어 아키텍처 설계서 (SW Architecture Specification) 5

1.1.1. 소프트웨어 시스템 환경 5

1.1.2. 소프트웨어 시스템 외부 인터페이스 5

1.1.3. 감정기반 표정/입 생성 서비스 API 호출 절차 8

**표 목차**

[표 1 외부 인터페이스 상호작용 7](#_Toc522213773)

[표 2 모듈 인터페이스 입/출력 명세 7](#_Toc522213774)

**그림 목차**

[그림 1 감정기반 표정/입 생성 모듈 외부 인터페이스 6](#_Toc522213775)

# SW 설계 사항

본 절에서는 감정기반 표정/입 생성 소프트웨어 혹은 시스템 설계 관점에서의 내용을 기술한다.

## 소프트웨어 아키텍처 설계서 (SW Architecture Specification)

시스템의 품질을 확보하기 위하여 전체시스템에 대한 청사진으로서의 아키텍처를 작성한다. 소프트웨어 아키텍처는 개발 대상 응용 소프트웨어에 대한 아키텍처이며, 시스템 아키텍처는 응용 소프트웨어와 이에 상호작용하는 환경 및 네트워크가 포함된 아키텍처를 의미한다. 본 절에서는 SW 시스템 환경, 외부 인터페이스, 모듈 구성도, 프로세스간 통신, 기능 흐름도를 기술한다.

### 소프트웨어 시스템 환경

감정기반 표정/입 생성 모듈

* 서버 측
  + Train된 모델 사용시 GPU 사용 여부 및 필요 RAM : 미정
  + 학습한 GPU : 미정
  + OS: Ubuntu Linux 18.04 LTS
  + Language: Python 3.6, C++14
  + Framework : Docker Server 18.06.1, Autodesk AnimX, libROSA 0.6.2, numpy 1.15.1
  + 서버 – 클라이언트 간 작동 : HTTP RESTful API 제공 - flask 1.0.2
* 모바일 측
  + OS: Android 8.0
  + Language: Java 8, Kotlin, C++
  + Framework : OpenGL ES
  + 서버 – 클라이언트 간 작동 : 안드로이드 앱 내에서 사용 가능한 Java 함수 제공

### 소프트웨어 시스템 외부 인터페이스

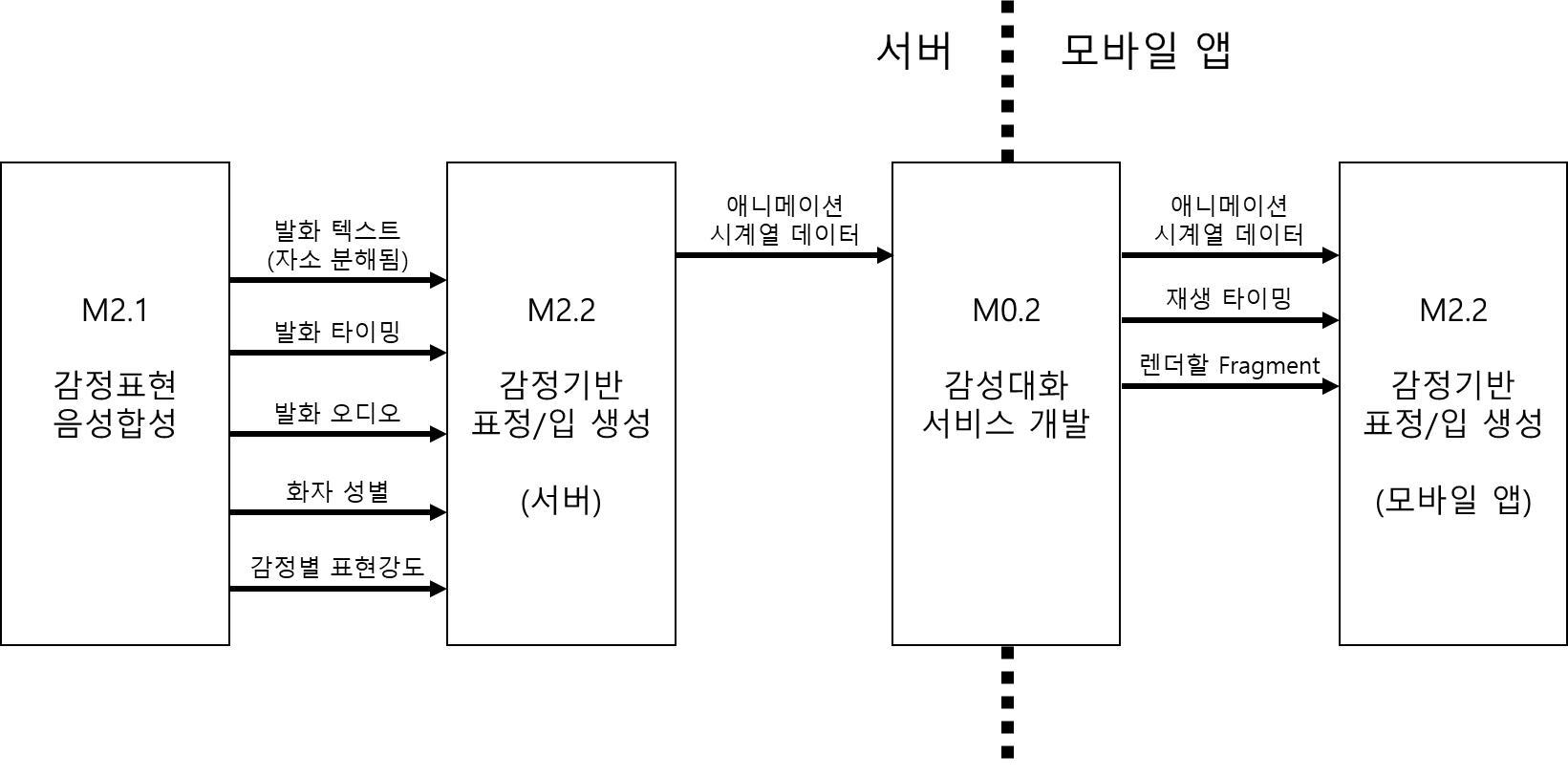


그림 1 감정기반 표정/입 생성 모듈 외부 인터페이스

표 1 외부 인터페이스 상호작용

|  |  |  |  |
| --- | --- | --- | --- |
| 시스템 설명 | 감정기반 표정/입 생성 모듈 (서버 측) | | |
| 상호작용 | 과제 내의 모듈 이름 | 모듈과의 상호작용 | 상세 설명 |
| M2.1 감정표현 음성합성 | Input | 음성합성기로 합성된 음성에 어울리는 표정 및 입모양 애니메이션 생성을 위해 음성 wav 파일, 발화 텍스트 txt, 음성합성 신경망 타이밍 정보 npy, 감정 종류 및 강도 인덱스, 성별 인덱스를 input으로 받음 |
| M0.2 감성대화서비스 개발 | Output | 생성된 애니메이션을 표상하는 시계열 정보를 output으로 제공 |

|  |  |  |  |
| --- | --- | --- | --- |
| 시스템 설명 | 감정기반 표정/입 생성 모듈 (모바일 측) | | |
| 상호작용 | 과제 내의 모듈 이름 | 모듈과의 상호작용 | 상세 설명 |
| M0.2 감성대화서비스 개발 | Input | 본 모듈 서버 측 API를 통해 생성된 애니메이션 시계열 정보 text를 input으로 받음 |
| M0.2 감성대화서비스 개발 | Output | 렌더된 애니메이션 화상을 output으로 제공 |

표 2 모듈 인터페이스 입/출력 명세

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 감정기반 표정/입 생성 모듈 (서버 측) | | | | |
| Input/Output | Name | Type | Description | Default |
| Input | sentence\_jaso | File (.txt) | 자소 분해된 발화 텍스트. (.txt 파일) | 없음 (필수 입력사항) |
| Input | sentence\_neural\_timing | File (.npy) | 발화 텍스트의 자소가 발음되는 타이밍 정보를 2D array로 입력 (.npy 파일) | 없음 (필수 입력사항) |
| Input | sentence\_audio\_wav | File (.wav) | 발화 텍스트를 음성합성한 오디오 Mono Waveform을 입력 (Sampling Rate: 8000Hz) | 없음 (필수 입력사항) |
| Input | speaker\_gender | Number | 화자 음성상의 성별.  성별은 서비스 코드 규약을 따름. | 없음 (필수 입력사항) |
| Input | emotion\_strength[10001] | Number | Happiness 감정의 표현 강도를 0~3 사이의 int 값으로 입력.  감정은 서비스 코드 규약을 따름. | 미입력시 0으로 간주 |
| Input | emotion\_strength[10002] | Number | Anger 감정의 표현 강도를 0~3 사이의 int 값으로 입력.  감정은 서비스 코드 규약을 따름. | 미입력시 0으로 간주 |
| Input | emotion\_strength[10003] | Number | Disgust 감정의 표현 강도를 0~3 사이의 int 값으로 입력.  감정은 서비스 코드 규약을 따름. | 미입력시 0으로 간주 |
| Input | emotion\_strength[10004] | Number | Fear 감정의 표현 강도를 0~3 사이의 int 값으로 입력.  감정은 서비스 코드 규약을 따름. | 미입력시 0으로 간주 |
| Input | emotion\_strength[10006] | Number | Sadness 감정의 표현 강도를 0~3 사이의 int 값으로 입력.  감정은 서비스 코드 규약을 따름. | 미입력시 0으로 간주 |
| Input | emotion\_strength[10007] | Number | Surprise 감정의 표현 강도를 0~3 사이의 int 값으로 입력.  감정은 서비스 코드 규약을 따름. | 미입력시 0으로 간주 |
| Output | animation\_data | File (.xml) | 본 모듈 모바일 측에서 처리 가능한 형태의 애니메이션 데이터 XML 파일  HTTP 응답으로 XML 파일의 내용을 반환함. |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 감정기반 표정/입 생성 모듈 (모바일 측) | | | | |
| Input/Output | Name | Type | Description | Default |
| Input | animation\_data | string | 본 모듈 서버 측에서 생성한 애니메이션 데이터 XML 파일의 텍스트 내용 | 없음 (필수 입력사항) |
| Input | time | int | 재생하고자 하는 애니메이션의 타이밍(재생 시점)을 밀리초 단위의 int 값으로 입력 | 없음 (필수 입력사항) |
| Input | target\_fragment | android.app.Fragment | 애니메이션을 렌더하게 될 Fragment 오브젝트 | 없음 (필수 입력사항) |

### 감정기반 표정/입 생성 서비스 API 호출 절차

감정기반 표정/입 생성 (서버 측) 서비스 API의 호출 순서는 다음과 같다.

1. flagship-server git 저장소를 clone한다.
2. 저장소 내 주어진 Dockerfile을 docker build -t emotional-facial-animiation-server . 명령으로 빌드한다.
3. 빌드된 Docker 이미지를 docker run -d -p 3456:5000 emotional-facial-animiation-server명령으로 실행한다.
4. 열린 3456 HTTP 포트의 “/emotional-facial-animation/rest/generate-animation” 주소를 대상으로 MIME 타입이 multipart/form-data인 POST 를 요청한다.
5. HTTP 응답으로 XML 데이터가 반환된다.

감정기반 표정/입 생성 (모바일 측) 서비스 API의 호출 순서는 다음과 같다.

1. flagship-android git 저장소를 clone한다.
2. 저장소 내 주어진 ‘vml-animation.aar’ 파일을 프로젝트에 포함한다.
3. 해당 패키지의 AvatarFragment를 Activity에 배치한다.
4. AvatarAnimation생성자로 위 Fragment를 넘겨 인스턴스를 생성한다.
5. AvatarAnimation.setAnimation() 함수를 호출한다. 이 때 파라미터로 XML 파일 데이터를 InputStream으로 전달한다.
6. 재생할 오디오 파일이 있을 시, MediaPlayer 인스턴스를 통해 오디오를 재생 시작한다.
7. MediaPlayer.getCurrentPosition() 을 호출하여 현재 오디오가 재생되는 타이밍을 얻는다.
8. 알아낸 타이밍을 파라미터로 하여 AvatarAnimation.updateAnimation() 함수를 호출하여 해당 타이밍에 알맞은 애니메이션을 재생한다.
9. 7번부터의 과정을 무한 루프하여 실시간으로 애니메이션을 재생한다.
10. 오디오가 끝나면, AvatarAnimation.playIdleMotion() 을 호출하여 기본 애니메이션을 재생한다.