소프트웨어실습

요구사항명세서

2017.12.09.

문범우

2013440043

차 례

[1. 소개(Introduction) 1](#_Toc224639880)

[1.1. 목표 1](#_Toc224639881)

[1.2. 범위 1](#_Toc224639882)

[1.3. 용어 및 정의](#_Toc224639883) 1

[1.4. 참고문헌 2](#_Toc224639885)

1.5 [개요 (Overview)..................................................................................................................2](#_개요_(Overview))

[2. 일반사항 3](#_Toc224639886)

[2.1. 타 시스템과의 관계](#_Toc224639887) 3

[2.2. 제약사항 10](#_Toc224639888)

[2.3. 가정 10](#_Toc224639889)

[2.4. 위험요소 11](#_Toc224639890)

[3. 요구사항 분석 12](#_Toc224639891)

3.1 기능 요구사항(Functional Requirements)………………………………………………………...…12

[3.1.1. 기능요구사항 1 12](#_Toc224639892)

[3.1.2. 기능요구사항 2](#_Toc224639893) 14

[3.2비기능 요구사항 (Nonfunctional Requirements)…………………………………………………...15](#_Toc224639894)

[3.2.1. 시스템 기본요건 15](#_Toc224639895)

[3.2.2. 기술요구사항 16](#_Toc224639896)

[3.2.3. 인터페이스요구사항 16](#_Toc224639897)

4. 시스템 설계……………………….……………………………………………………………………………..18

5. 시스템 구현……………………………………………………………………………………………………..21

[6. Appendix 26](#_Toc224639898)

**초록(Abstract)**

본 논문에서는 긴급상황 및 경보 등에서 사용되는 AEAT를 1:1방식 또는 1:다 방식으로 메세지를 송수신하기 위해 설계 및 구현된 소프트웨어를 서술한다. 또한 AEAT형식을 XML schema로 저장하거나 파일로 저장된 XML schema를 AEAT형식으로 불러와 사용자 및 관리자가 보다 빠르고 간편하게 또는 작성 및 수정에 용이하도록 한다.

# 소개(Introduction)

## 목표 (Purpose)

a) AEAT(Advance Emergency Alert Table) 프로그램을 구현한다.

b)입력 내용을 Marshalling 하여 XML파일을 만들어 저장한다.

c) XML파일을 UnMarshalling 한다.

d) AcitveMQ 및 JMS를 이용하여 메시지를 송수신한다.

e) Java Swing에 대하여 학습하고 사용자편의 GUI를 구현한다.

## 범위 (Scope)

1. 해당 소프트웨어는 AEAT Editor 이다.
2. 해당 소프트웨어를 통해 특정 사용자에게 긴급한 상황에 대한 메시지를 송신할 수 있다.
3. 해당 소프트웨어를 통해 특정 사용자는 긴급한 상황에 대한 메시지를 수신할 수 있다.
4. 프로그램 사용자는 AEAT양식을 알맞게 작성하여 XML파일을 생성할 수 있다.
5. 프로그램 사용자는 ActiveMQ및 JMS를 통해 메시지를 송수신 할 수 있다.
6. 프로그램 사용자는 송수신된 메시지를 AEAT Editor를 통해 편집할 수 있다.
7. 프로그램 사용자는 프로그램에서 메시지 송수신 목적지를 설정할 수 있다.
8. 해당 소프트웨어는 AEAT 양식에 맞추어 필요한 상황을 입력하고, 이를 송수신하거나 XML파일로 save 및 load한다.

## 용어정의 및 약어 (Definition, acronyms, and abbreviations)

AEAT: Advance Emergency Alert Table

JMS: Java Message Service

## 참고문헌

<http://mmlab.uos.ac.kr/Capstone/Default.htm>

XML Schema

<https://www.atsc.org/wp-content/uploads/2016/01/S33-331r0-WD-PS-Signaling-Deivery-Sync-FEC-S33-1-788r5.pdf>

## 개요 (Overview)

1. 해당 소프트웨어는 위급상황을 알리기 위한 테이블을 편집하거나 이를 JMS로 송수신할수 있도록 만들어진 것이다.
2. 사용자의 편의성을 위해 Radio Button과 Combo Box 및 Tabbed Pane을 활용하였다.
3. 해당 소프트웨어는 AEAT main에서 프로그램 실행과 버튼에 대한 기능설정을 진행하고, AEAT network에서 JMS와 ActiveMQ를 이용한 메시지 송수신 기능을 수행하며 AEAT view에서 프로그램의 GUI를 구성되도록 설계되었다.

# 일반사항 (Overall description)

## 타 시스템과의 관계 (Product Perspective)

### 시스템 인터페이스(System interface)

해당 소프트웨어는 아래 UModel 그림과 같이 network(JMS)와 view(AEAT Editor), aeat가 연결되어 있다.

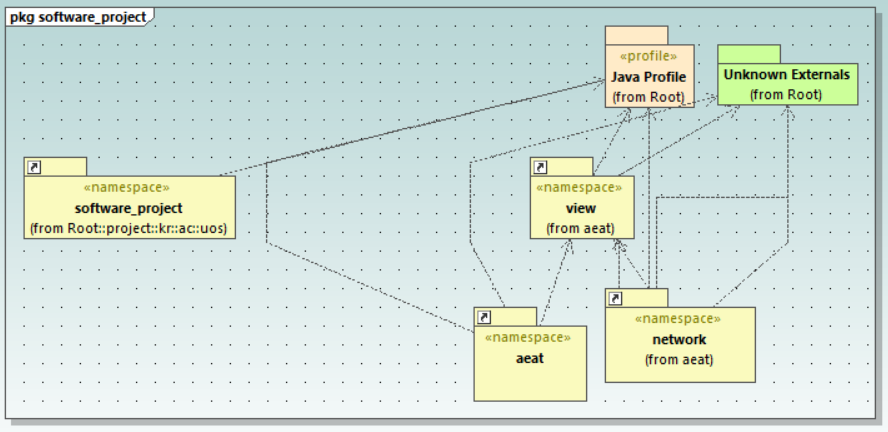


Figure Overall UModel

사용자가 network를 통해 수신한 메세지는 view로써 표현되고, 이때 aeat양식을 따른다.

또한 view에서 aeat 양식에 따라 작성한 것을 network를 통해 수신할 수 있다.

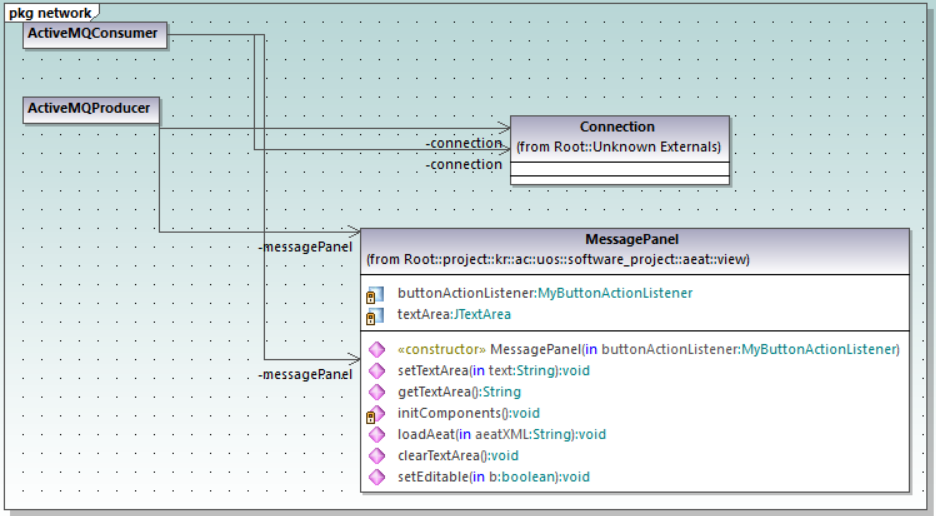


Figure Network UModel

Figure 2를 통해 Network UModel을 확인할 수 있다. ActiveMQConsumer와 ActiveMQProducer가 함께 Connection 및 MessagePanel과 연관되어 있다.

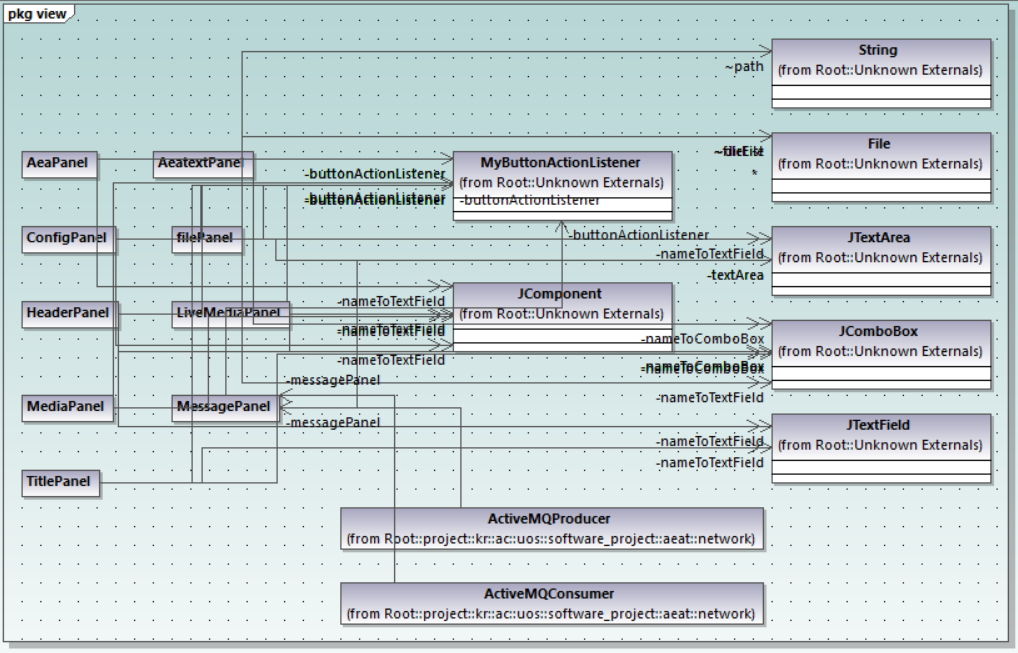


Figure View UModel

Figure 3을 통해 View의 UModel을 확인할 수 있다.

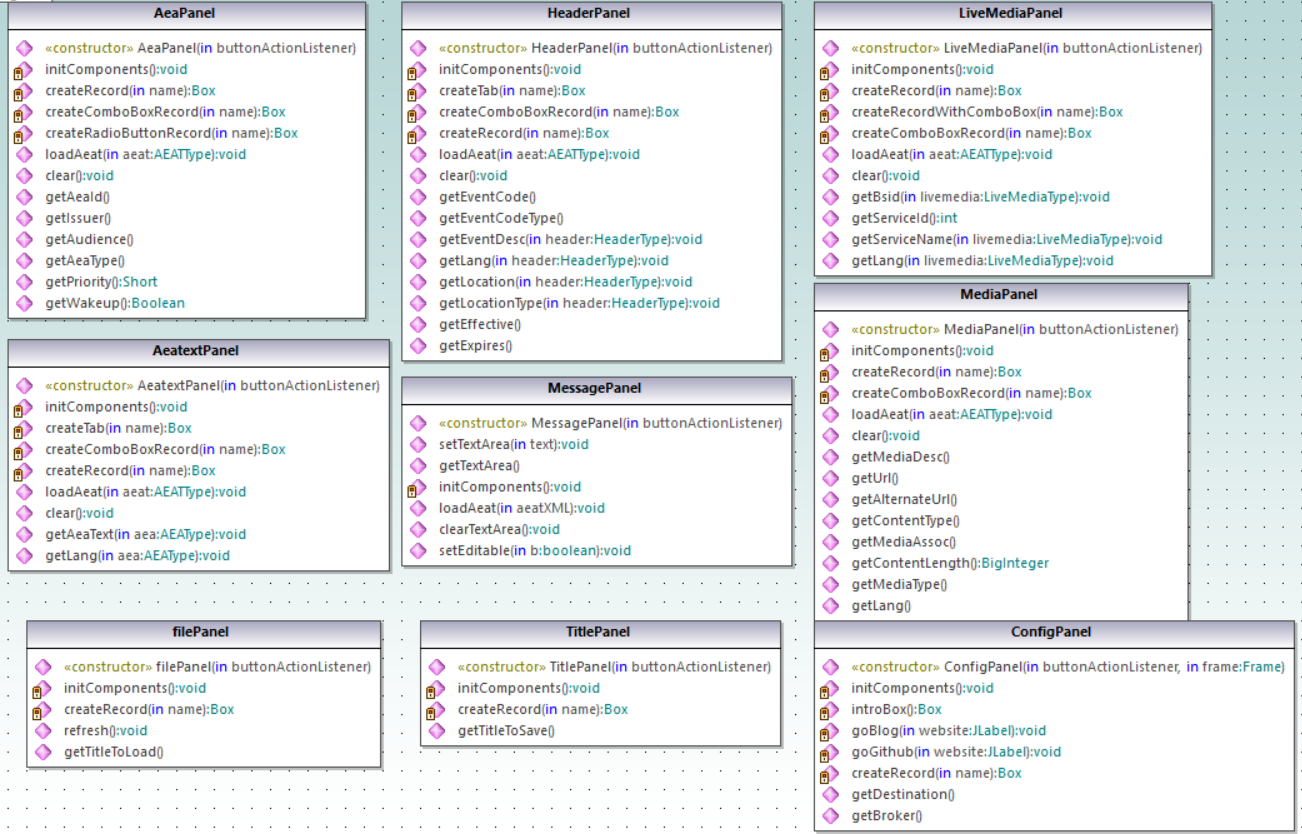


Figure Interface of each Panel

Figure 4에서 각 패널에 대한 인터페이스를 확인할 수 있다.

AEAT Editor는 각 Panel을 통해 구성된다. 각 Panel에서는 이후 Marshalling 또는 UnMarshalling을 위한 기능을 수행한다. AEAT Editor는 view에서 구현되며, view는 network와 연결되어 메시지 송수신이 가능하다.

### 사용자 인터페이스(User interfaces)

사용자는 해당 소프트웨어를 실행하였을 때 다음과 같은 화면을 볼 수 있다.

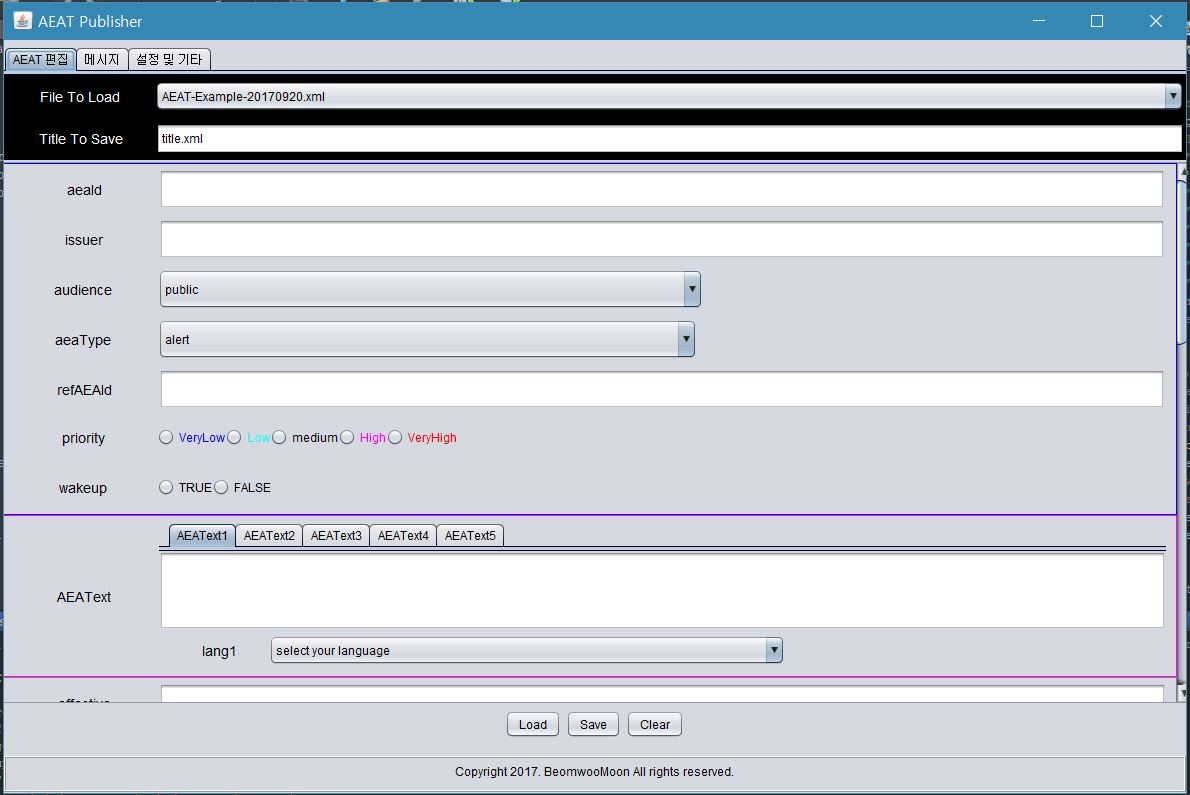


Figure AEAT Editor

Figure 5는 AEAT Editor 화면이다. 사용자는 해당 화면에서 xml파일을 불러와 UnMarshalling하여 화면의 구성에 맞도록 AEAT를 확인할 수 있다. 또한 해당 내용을 편집하거나 새로 작성하여 Marshalling 과정을 통해XML파일로 저장할 수 있다.

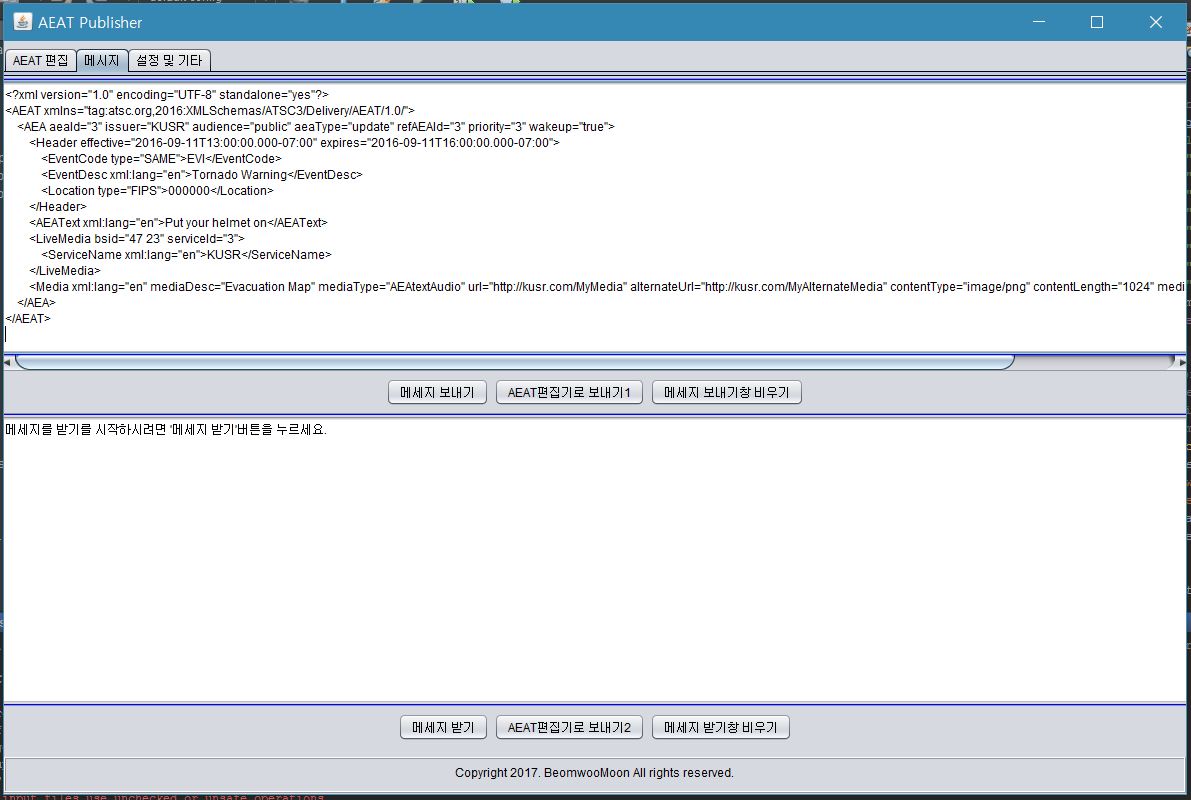


Figure Message View

Figure 6를 통해 메시지 view를 확인할 수 있다. 사용자는 앞의 AEAT Editor를 통해 불러온 XML파일을 바로 메시지 송신창에 load되는 것을 확인할 수 있다. 전체적인 view는 위에서 언급한 바와 같이 network와 연결되어 있어서 메시지 송수신이 가능하다. 메시지 보내기버튼을 통해서 현재 송신창에 있는 텍스트를 지정된 곳으로 송신할 수 있다. 또한 수신창에서 메시지 받기 버튼을 통해 지정된 곳으로부터 텍스트를 수신할 수 있다. 메시지 송수신창 어디에서든지 해당 텍스트를 AEAT Editor로 전달할 수 있다.

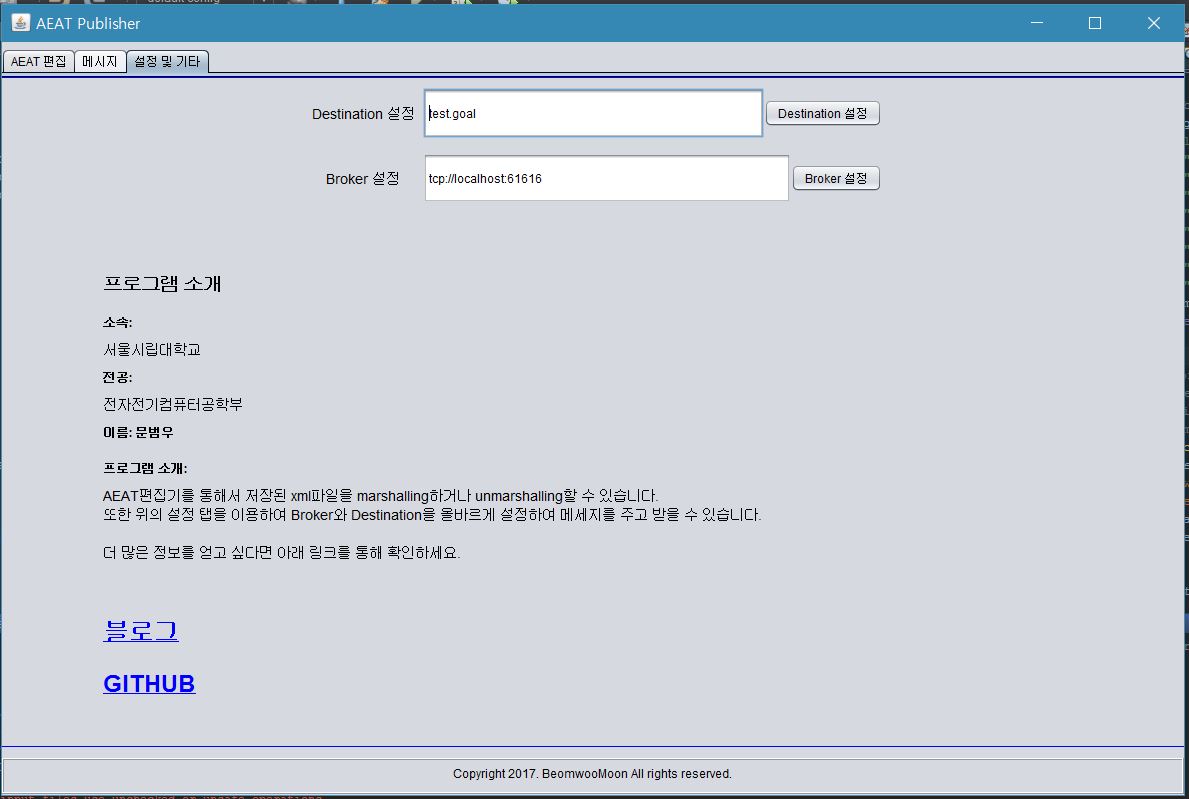


Figure Config view

사용자는 위의 그림 Figure 7의 Config view에서 Destination 및 Broker를 직접 설정할 수 있다. 또한 프로그램에 대한 간단한 설명과 개발자의 신상이 소개되어 있다.

### 운영(Operations)

1. Load XML schema

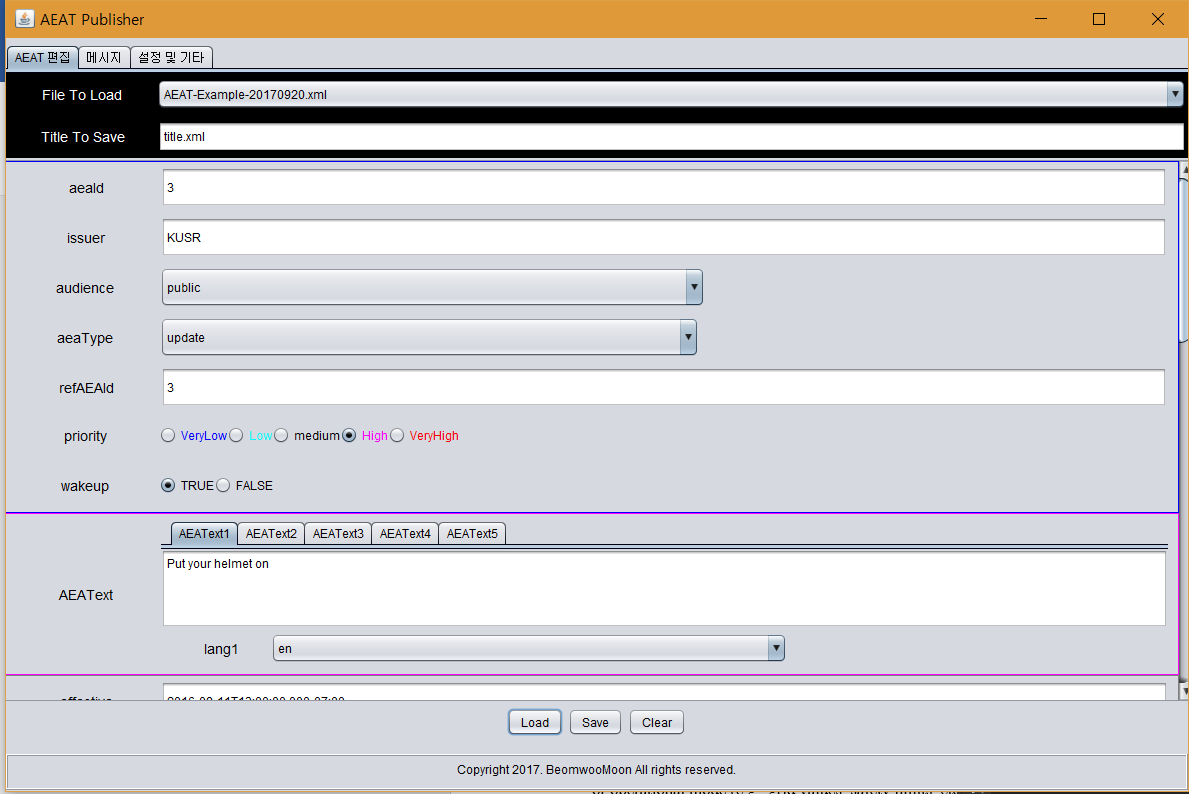


Figure Load XML schema

사용자는 Figure 8 에서 볼 수 있는 File To Load와 Load버튼을 이용하여 xml파일로 저장되어 있는 파일을 UnMarshalling 하여 AEAT 양식에 맞춰서 확인할 수 있다.

File To Load는 “xml/”경로에 있는 파일을 불러온다.

1. Save AEAT to XML schema

Figure 8을 보면 Title To Save와 Save버튼을 확인할 수 있다. 사용자는 현재 작성된 AEAT를 XML schema로 저장하기 위하여 저장될 파일의 이름을 Title to Save에서 지정할 수 있으며 Save버튼을 누르면 해당 이름으로 파일이 저장된다. 파일은 “xml/”경로로 저장되며, 저장이 됨과 동시에 파일목록을 새로 읽어와서 File To Load에 있는 항목이 업데이트 되기 때문에 새롭게 저장된 xml파일을 실시간으로 확인할 수 있다.

1. Send message through ActiveMQ

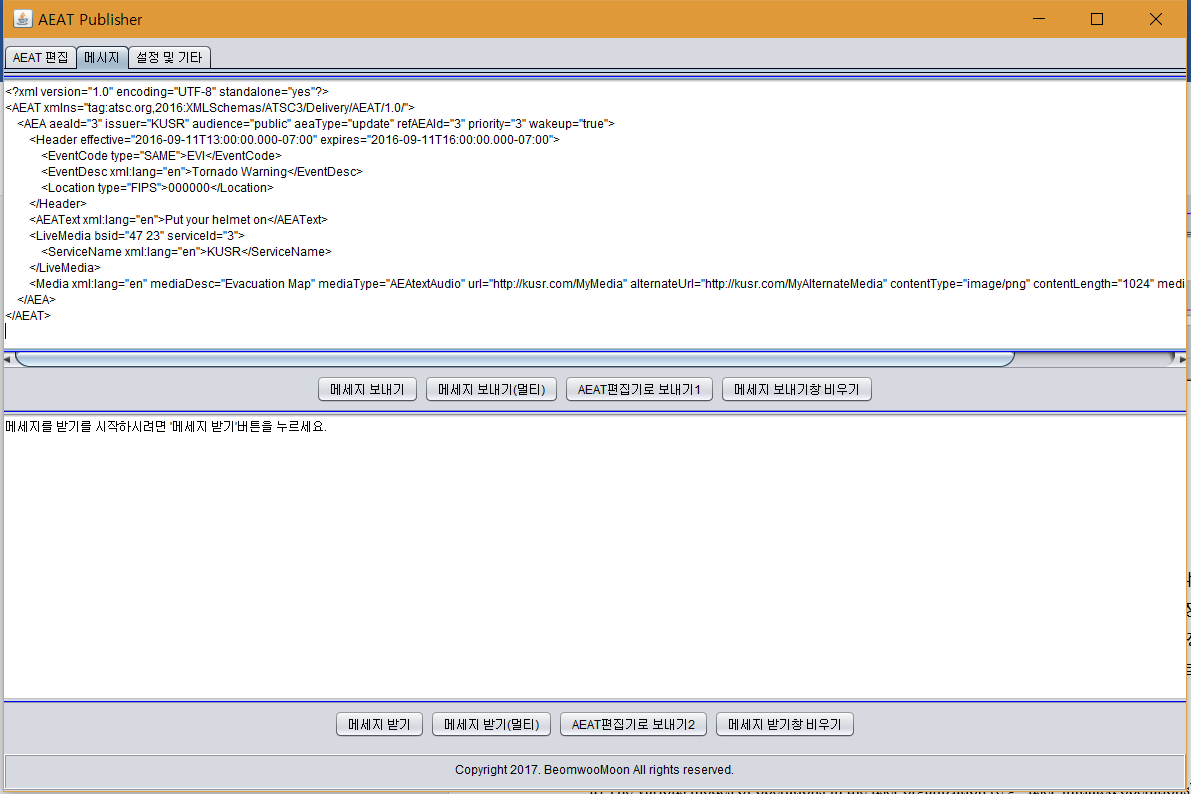


Figure Message view

사용자는 위의 Figure 9 에 보이는 Message view에서 ActiveMQ를 이용한 메시지 송수신이 가능하다. 1:1방식으로 메시지를 송수신하기 위해서는 메시지 보내기 및 메시지 받기 버튼을 누르면 되며, 1:다 방식으로 메시지를 송수신하기 위해서는 메시지 보내기(멀티) 및 메시지 받기(멀티)를 누르면 된다.

메시지를 보내면 상단의 입력창의 텍스트가 사라지면서 지정된 Destination과 Broker로 메세지가 송신된다.

1. Receive message through ActiveMQ

사용자는 Figure 9에 보이는 메시지 받기 또는 메시지 받기(멀티) 버튼을 통해 메시지를 수신할 수 있다. 메시지 받기를 시작하면 하단의 수신창에 메시지를 기다리는 중이라는 메세지를 확인할 수 있으며, 다른 사용자가 지정된 Destination 및 Broker로 메시지를 송신하면 이를 받아서 수신창에 띄어준다.

## 제약사항 (Product functions)

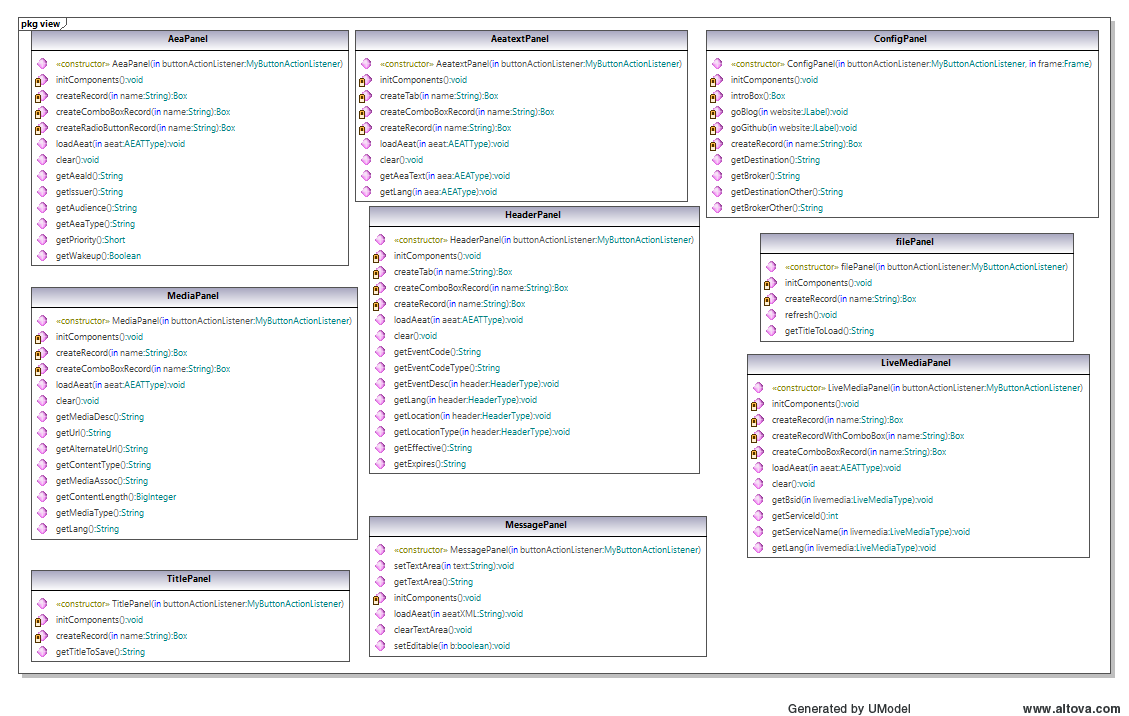


Figure Function

Figure 10을 통해 전체적인 Function을 확인할 수 있다.

1. 사용자는 직접 AEAT를 작성하여 XML schema로 저장할 수 있다.
2. 사용자는 XML schema 파일을 불러와 AEAT 양식에 맞게 확인할 수 있다.
3. 사용자는 AEAT를 다른 사용자에게 송신할 수 있다.
4. 사용자는 AEAT를 다른 사용자로부터 수신할 수 있다.

## 가정 (User characteristics)

해당 소프트웨어 관리자 및 사용자는 특정 보안 및 기상관련 관제센트의 관리자 및 알림관련 부서 관리자 또는 그 산하 부서 및 직원이라고 가정한다.

사용자는 해당 소프트웨어를 통해 신속, 긴급하게 1:다 방식으로 송신해야 하는 긴급 재난관련 메시지 및 기상 메시지 또는 기타 긴급성이 필요로 되는 메시지를 XML schema로 송신이 가능하다. 직접적으로 XML schema에 맞춰서 메시지를 작성하지않고 간단하게 해당 소프트웨어의 AEAT 편집기를 이용하여 필요한 메시지를 작성할 수 있으며 이를 XML schema로 저장하여 추후 필요한 경우 확인할 수 있다.

사용자는 추가적으로 1:1 방식으로도 메시지 송신 및 수신이 가능하다.

## 위험요소 (Constraints)

해당 소프트웨어 개발자는 현재 위에서 언급한 내용들을 준수하여 프로그램을 설계 및 개발하며 아래의 사항들을 준수한다.

* 프로그램의 추가적인 기능에 대해서는 주석처리를 통해 설명한다.
* 프로그램의 각각의 기능들이 독립적으로 수행되며, 필요시 유동적으로 연관을 갖는다.
* 사용자에게 보다 친숙하고 사용하기 편리한 시스템 설계를 진행한다.
* 추후 추가적인 시스템 개발과 확장성 및 가용성을 고려한다.

해당 소프트웨어 사용자 및 관리자는 다음과 같은 위험요소를 고려하고 준수한다.

* 해당 소프트웨어를 통해 불법적인 업무를 수행하거나 악의를 갖고 프로그램을 수행하지 않는다.
* 해당 소프트웨어에 대한 모든 권한은 저자에게 있음을 표시한다.
* 프로그램에 대한 추가적인 유지보수가 필요할 시 저자를 통해 진행된다.
* 프로그램에 대한 임의적인 변경 및 일부 항목 또는 기능을 삭제하지 않는다.

## 가정 및 종속성 (Assumptions and dependencies)

본 문서에서 언급한 소프트웨어의 모든 구현은 java를 통해 구현되었다. 또한 기본적인 AEAT양식을 위하여 공식홈페이지를 이용하여 이를 구현하였다. 그리고 메시지 송수신을 위하여 ActiveMQ를 이용하였다. 메시지 송수신의 경우 개인 사용자는 메시지를 자신의 서버에서 송수신하기 위해 필요한 61616포트를 개방해야 한다.

# 요구사항 분석

## 기능 요구사항 (Functional Requirements)

해당 소프트웨어는 기능적으로 AEAT 구현과 ActiveMQ를 이용한 메시지 송수신을 요구한다. 추가적으로 AEAT를 사용자가 작성 및 편집하기 용이하도록 RadioButton과 ComboBox 및 TabbedPane의 적절한 사용을 요구한다.

이에 따라서 본 절에서는 AEAT 구현을 하기 위해 AEAT Type의 전체적인 구조를 살펴보고 이후 메시지 송수신의 구조와 기능수행에 대해서 이야기한다.

### 기능요구사항 1: AEAT 구현

본 절에서는 AEAT를 구현하기 위해 AEAT의 전체적인 구조를 살펴본다. 아래 트리는 XMLspy를 통해서 도출하였다.

1. AEAT Type

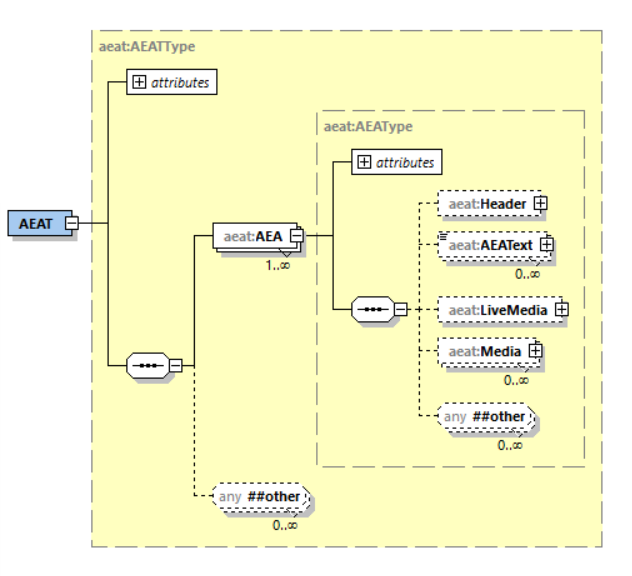


Figure Original AEAT Type schema

해당 소프트웨어에서는 Figure 11과 같은 AEAT Type schema를 편집하거나 저장 및 불러오기가 가능해야 한다. 각각에 대한 필요한 형식 및 구조는 다음페이지에 첨부된 Figure 12와 같다.

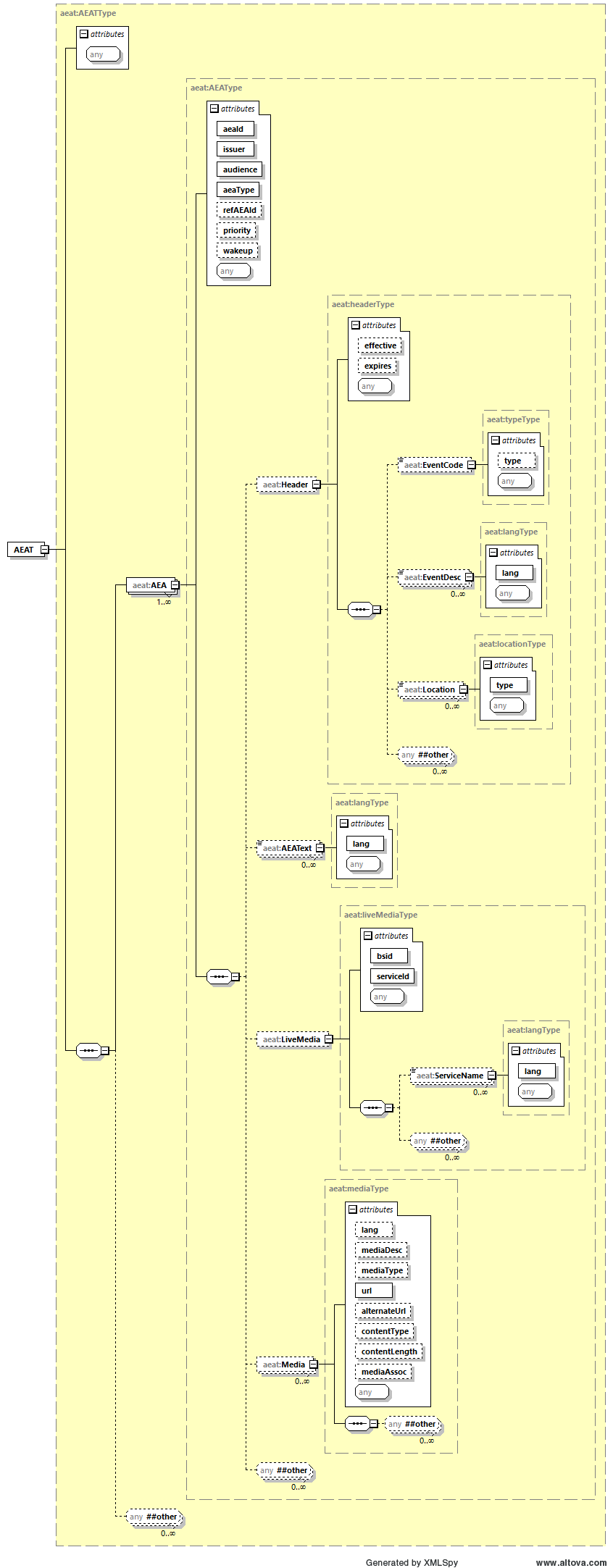


Figure Overall AEAT Type

### 기능요구사항 2: 메세지 송수신

메시지 송수신을 구현하기 위해 ActiveMQ를 활용한다.

ActiveMQ를 통해 구현되는 메시지 송수신 기능의 전체적인 도식도는 다음 Figure 13과 같다.

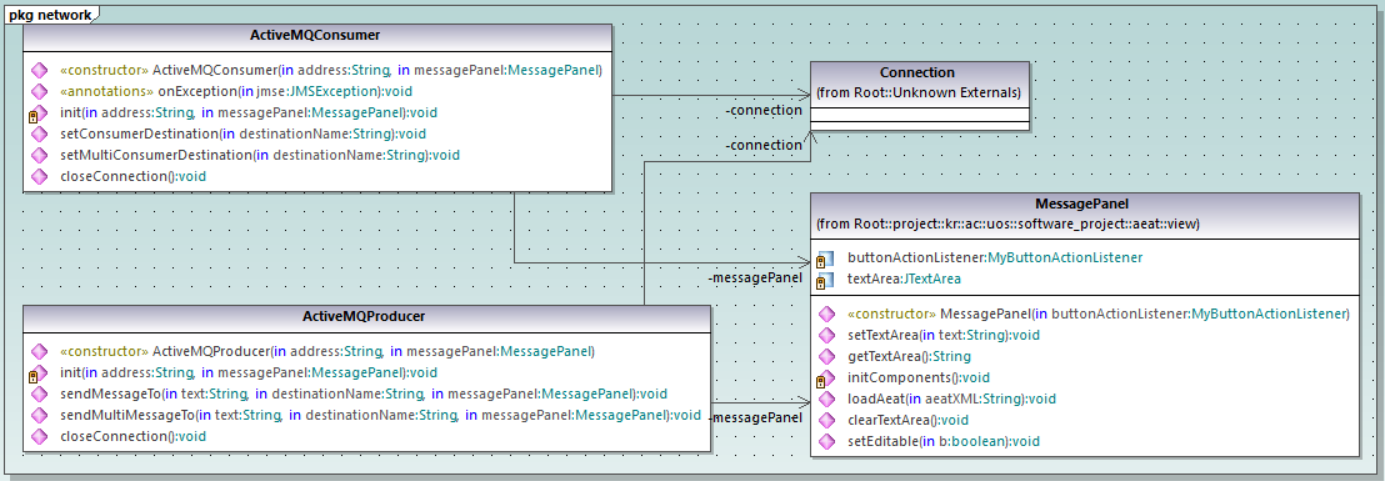


Figure Overall of Message Function

ActiveMQConsumer와 ActiveMQProducer가 함께 Connection과 MessagePanel과 연관되어 있다. 메시지를 송신하기 위해 ActiveMQProducer가 Connection을 통해 지정된 서버와 연결이된다. 이후 MessagePanel에 입력된 text를 송신한다. 비슷하게, 메시지를 수신하기 위해 ActiveMQConsumer가 Connection을 통해 지정된 서버와 연결이 된다. 이후 MessagePanel에 수신된 text를 출력한다.

## 비기능 요구사항 (Nonfunctional Requirements)

### 시스템 기본요건

#### 성능

해당 소프트웨어는 사용자가 작성 및 수정한 AEAT가 지정된 서버로 안전하게 전송되고 또한 지정된 서버로부터 AEAT 또는 추가적인 메시지를 수신할 수 있어야 한다.

메시지 송수신은 1:1 방식과 1:다 방식을 구현하여 필요한 기능에 따라 사용자가 선택할 수 있도록 한다.

#### 신뢰성

사용자 및 관리자가 해당 소프트웨어를 통해 AEAT 형식을 XML schema로 저장하거나 이를 불러들일 때 내용에 대한 손상이 없도록 하여 자체적인 AEAT 관리 유지에 있어서 신뢰성을 준다. 또한 메시지를 송수신할 때 사용자 및 관리자가 의도한대로 메시지가 송수신되며 text 및 data에 있어서 손상이 없도록 한다.

#### 유지보수성

해당 소프트웨어에 대한 유지보수 및 추가적인 개발이 진행될 때 설계되어있는 전체적인 구조에 대한 설명과 각 기능을 나타내는 함수에 대한 주석처리가 되어있음을 통해 유지보수가 용이하도록 한다.

#### 복구성

해당 소프트웨어에 대한 유지보수 작업 및 추가개발 작업이 진행되면서 일부 중요기능에 대한 처리가 되지 않거나 데이터 손상이 일어났을 때 이를 복구하기 위하여 github을 이용한다. Github을 통한 소프트웨어 프로젝트를 관리하며 추가적인 개발 및 유지보수가 진행되기 전의 상태를 보존함으로써 예기치 못한 상황에 대한 복구를 용이하게 한다.

#### 사용자인터페이스

사용자 및 관리자가 간단한 설명을 통해 해당 소프트웨어를 의도와 맞게 이용할 수 있도록 사용자 기반 UI를 설계한다. 각 버튼과 입력창 및 편집창에서 사용자가 자신의 의도한 기능을 수행할 수 있도록 적절한 기능 배치를 구현한다.

### 기술요구사항

#### 소프트웨어

* 해당 소프트웨어를 설계 및 개발하기 위해 Java Programming 능력을 필요로 한다.
* AEAT 형식을 다루기 위해 Marshalling 및 UnMarshalling에 대한 기초 개념을 필요로한다.
* 메시지 송수신을 구현하기 위해 ActiveMQ에 대한 사용법을 필요로 한다.

### 인터페이스요구사항

#### 인터페이스 기능

* AEAT를 편집할 수 있는 인터페이스가 존재하여 사용자 및 관리자가 간편하게 AEAT를 편집한다.
* 수정 및 보완된 AEAT가 XML schema로 저장될 수 있어 관리 및 유지보수가 용이하도록 한다.
* 과거의 XML schema를 불러와 AEAT형식으로 볼 수 있는 인터페이스가 존재한다.
* 사용자가 작성한 AEAT 또는 불러온 AEAT를 제 3자에게 송신할 수 있다.
* 제 3자로부터 AEAT형식 및 text를 수신할 수 있다.
* 제 3자로부터 수신된 AEAT 형식의 XML schema를 AEAT 편집기로 전달할 수 있다.

#### 전달하는 데이터 유형 및 포맷

* 사용자 및 관리자가 ActiveMQ를 통해 송신하는 메시지 형식은 크게 XML schema와 text이다. 그 중 XML schema는 아래 Figure 14와 같다.

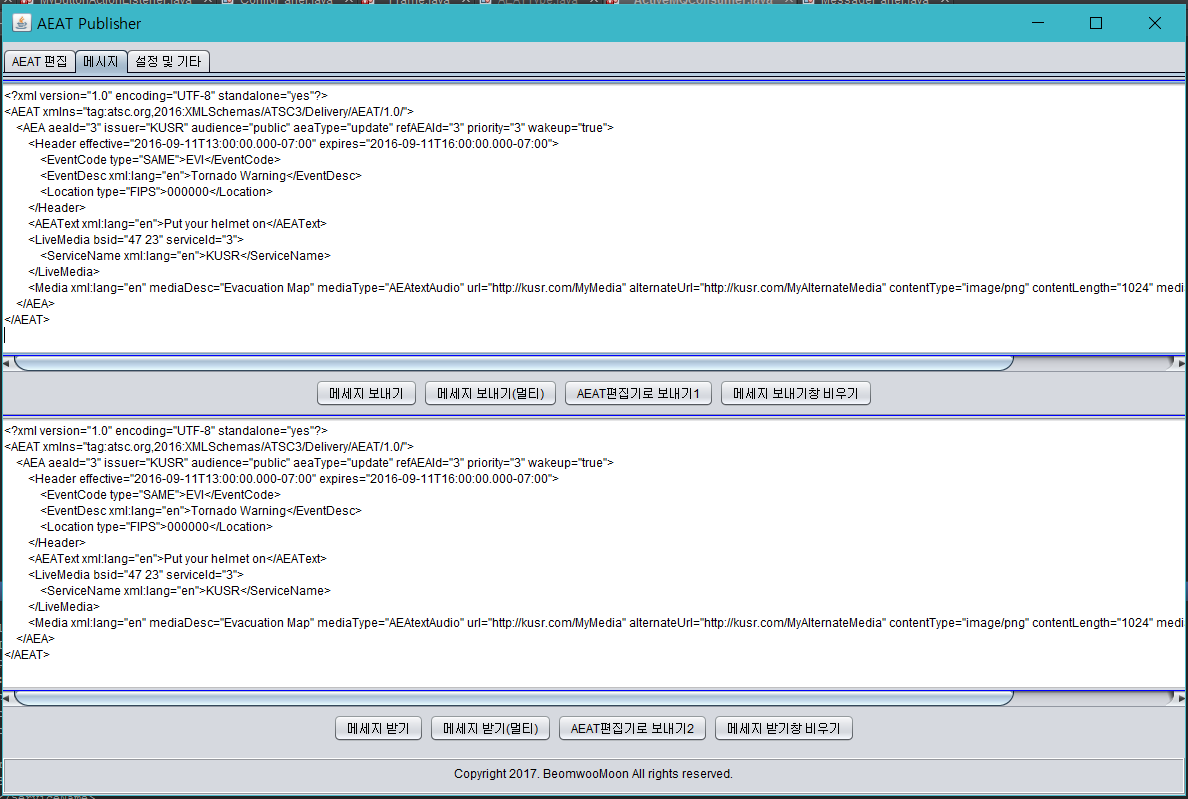


Figure Format of sending message

#### 전달받는 데이터 유형 및 포맷

* 사용자 및 관리자가 ActiveMQ를 통해 수신하는 메시지 형식은 크게 XML schema와 text이다. 그 중 XML schema는 아래 Figure 15와 같다

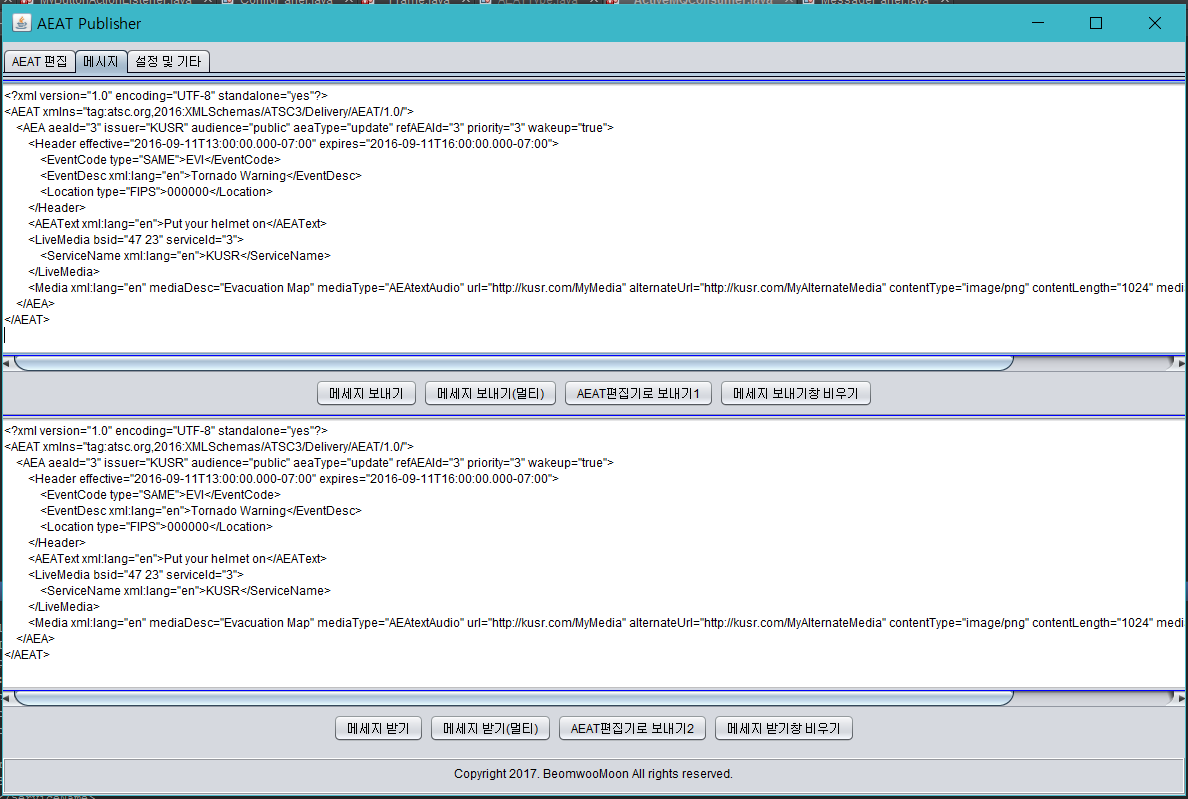


Figure Format of receiving message

# 시스템 설계

## GUI

* AEAT 편집기, 메시지, 설정 및 기타의 항목을 가진 총 3개의 TabbedPane을 구현한다.
* AEAT 편집기 내부에서는, AEAT 형식에 알맞게 작성할 수 있도록 항목을 구성한다.
* AEAT편집기 내부에서 특정 값을 갖는 항목에 대해서는 RadioButton 또는 ComboBox를 이용한다.
* AEAT 편집기 내부에서 하나 이상의 값을 갖는 항목에 대해서는 TabbedPane을 이용한다.
* AEAT 편집기에서는 작성된 AEAT를 XML schema로 저장될 수 있도록 파일 이름을 작성하는 항목과 저장 버튼이 구현된다.
* AEAT 편집기에서는 XML schema를 불러 올 수 있도록 특정 경로에 있는 파일을 불러오는 항목을 구현한다.
* 메시지 창에서는 메시지를 송신할 입력창과 수신된 메시지가 표시될 수신창을 구현한다.
* 각각의 창에 대해 기능을 수행할 수 있는 버튼을 구현한다.
* 설정 및 기타 창에서는 사용자가 직접 Destination과 Broker를 설정할 수 있도록 구현된다.
* 설정 및 기타 창에서는 프로그램에 대한 간략한 소개와 개발자 소개가 포함된다.

## MODEL

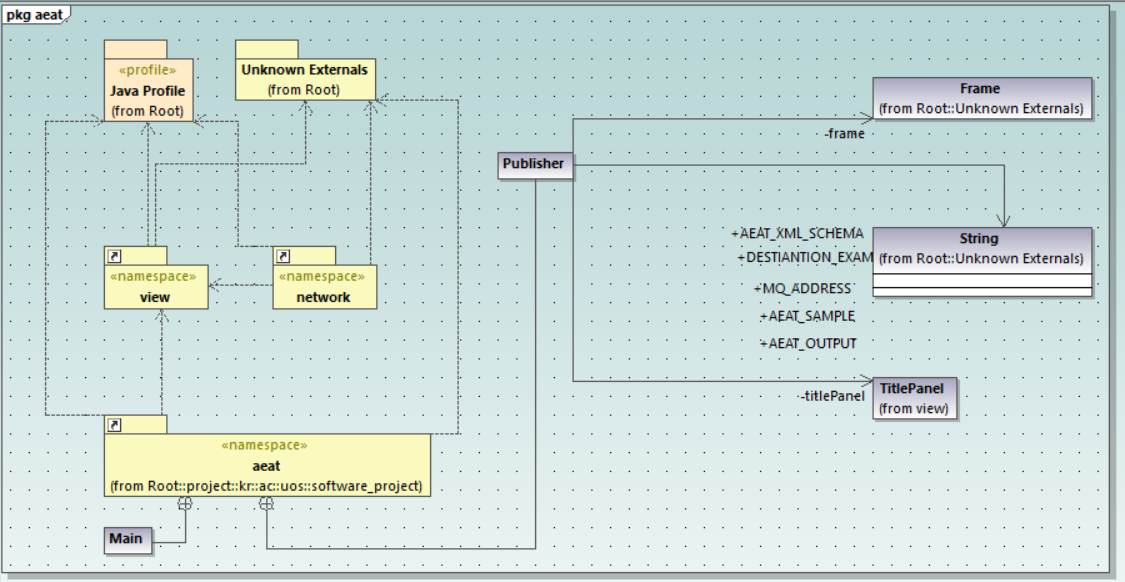


Figure Overall MODEL

본 문서에서 언급되는 소프트웨어의 전체적인 MODEL은 위의 Figure 16과 같다.

View와 network, aeat가 서로 연관되어 있으며 이들이 유기적으로 데이터를 교환한다.

## Marshalling/Unmarshalling

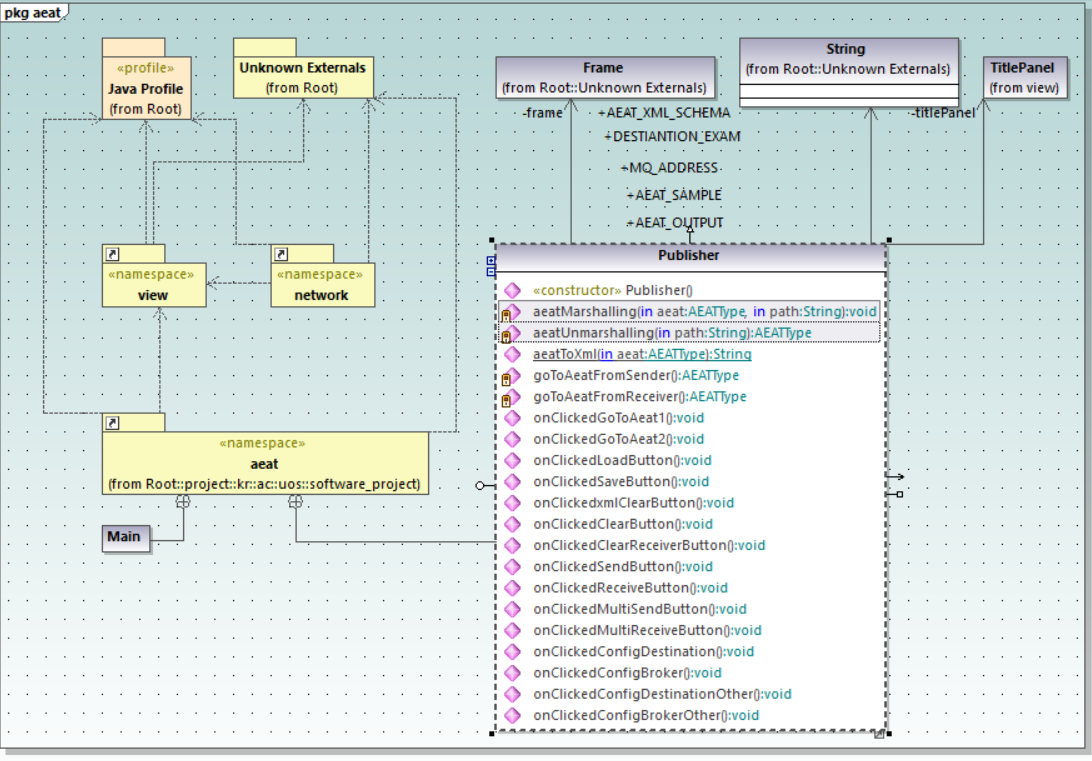


Figure Marshalling/Unmarshalling

Figure 17을 통해 publisher에서 구현되는 Marshalling/Unmarshalling을 확인할 수 있다.

## JMS Interface

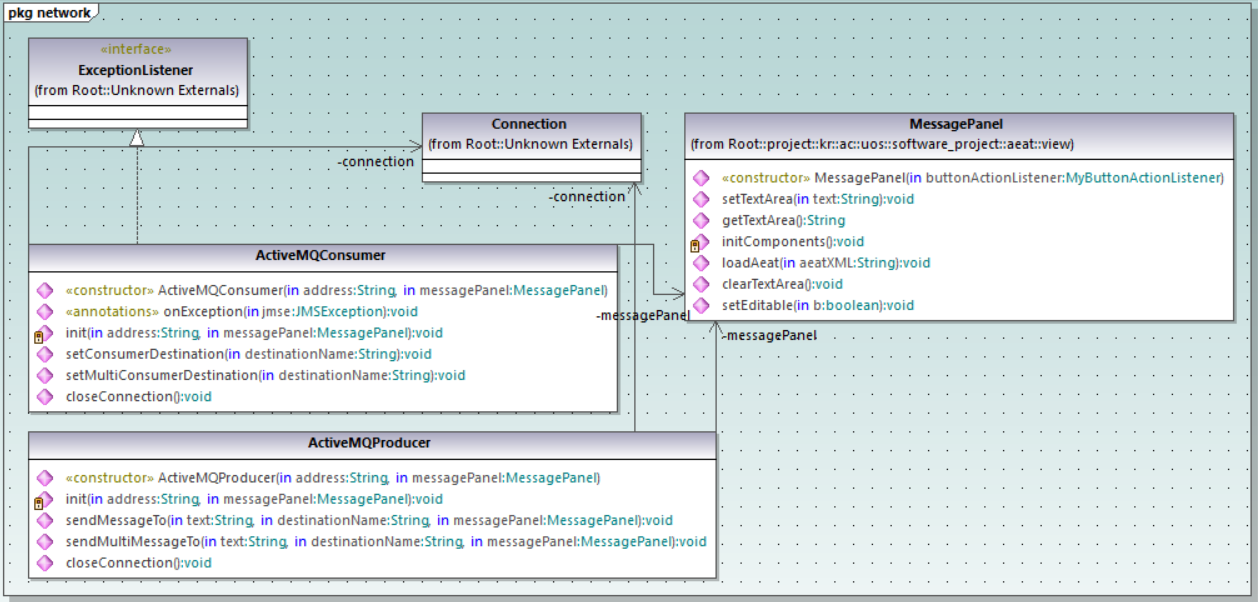


Figure JMS Interface through ActiveMQ

Figure 18에서 확인할 수 있듯이 JMS는 ActiveMQ를 통해 구현된다. 메시지를 송신하기 위해 ActiveMQProducer에서 Connection이 이루어지며 sendMessageTo를 통해 1:1 방식의 메시지 송신이 가능하며 sendMultiMessageTo를 통해 1:다 방식의 메시지 송신이 가능한다. 또한 ActiveMQConsumer에서 Connection이 이루어지며 setConsumerDestination을 통해 1:1 방식의 메시지 수신이 이루어지며, setMultiConsumerDestination을 통해 1:다 방식의 메시지 수신이 이루어진다.

# 시스템 구현

## GUI

아래 Figure 19,20,21 을 통해 구현된 소프트웨어의 GUI를 확인할 수 있다.

Figure 19를 확인하면, 상단에 File To Load와 Title To Save를 통해 AEAT를 XML schema로 저장하거나 저장된 XML schema를 AEAT로 불러올 수 있다.

또한 각 항목에 대해서 작성 및 수정이 용이하도록 항목이 구성되었으며 audience 및 aeaType등 특정 값으로 정해져 있는 항목은 ComboBox를 활용하거나 priority와 같이 RadioButton을 이용하였다. 그리고 AEAText와 같이 여러 개의 내용을 포함할 수 있는 항목은 그 내부에서 TabbedPane을 가지도록 구현하였다.

Figure 20을 확인하면, 메시지 송수신을 위한 두개의 패널과 각각의 버튼을 확인할 수 있다. 상단이 메시지를 송신하기 위한 텍스트를 입력하는 창이며 하단이 수신된 메시지가 출력될 창이다. 각각의 버튼에서 1:1 방식의 메시지 송수신 또는 1:다 방식의 메세지 송수신을 사용자가 선택할 수 있다. 또한 각 창에 있는 내용을 Figure 19의 AEAT 편집기로 전달할 수 있도록 버튼이 구현되어 있다. 추가적으로 창의 내용을 비우기 위한 버튼이 구현되어 있다.

Figure 21을 확인하면 Destination과 Broker를 설정할 수 있는 설정창이 구현되었다. 메시지메 올바르게 송수신하기 위해 2개의 Destination과 2개의 Broker를 설정할 수 있도록 구현하였고 이를 설정하면 console창에 설정이 적용되었다는 출력이 표시된다. 또한 하단에는 간단한 프로그램 소개와 개발자에 대한 소개가 나와 있다.

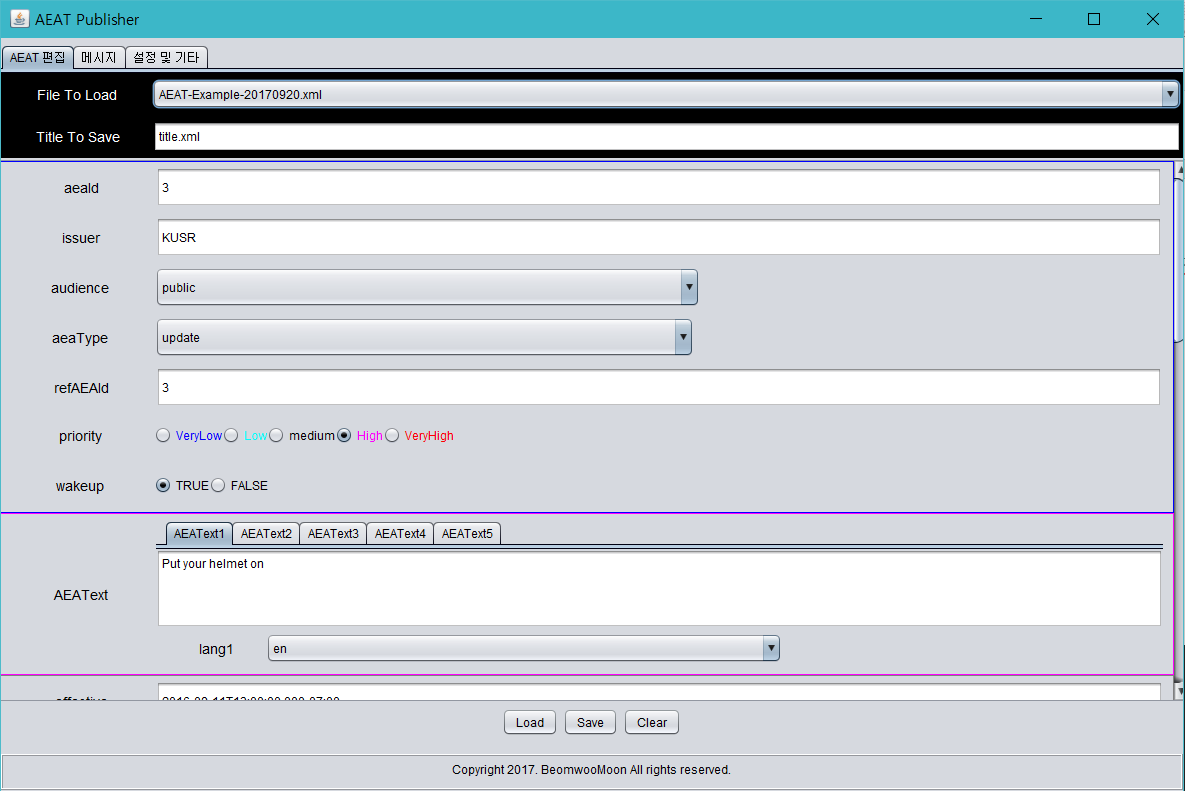


Figure AEAT Editor

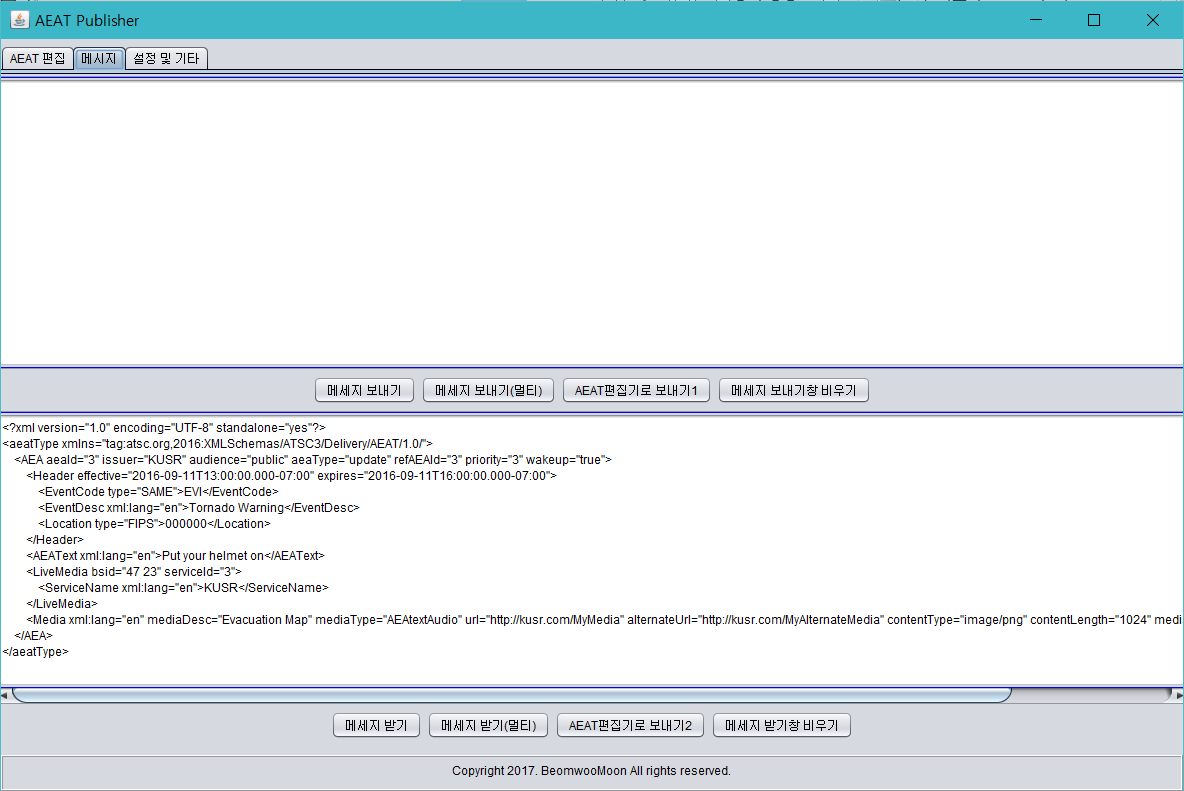


Figure Message send or receive

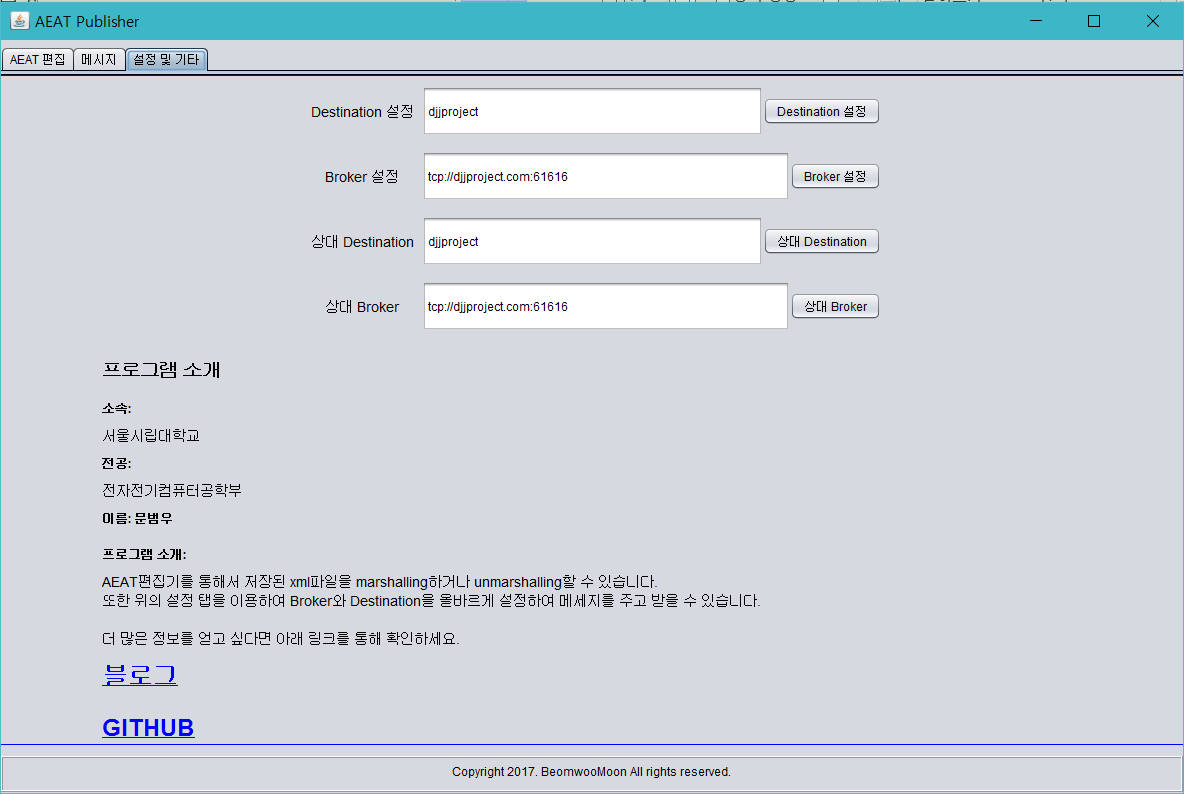


Figure Config tab

## MODEL

구현된 소프트웨어의 MODEL은 4.2에서 소개한바와 같이 설계되었다.

View와 network, aeat가 서로 연관되어 있으며 이들이 유기적으로 데이터를 교환한다.

## Marshalling/Unmarshalling

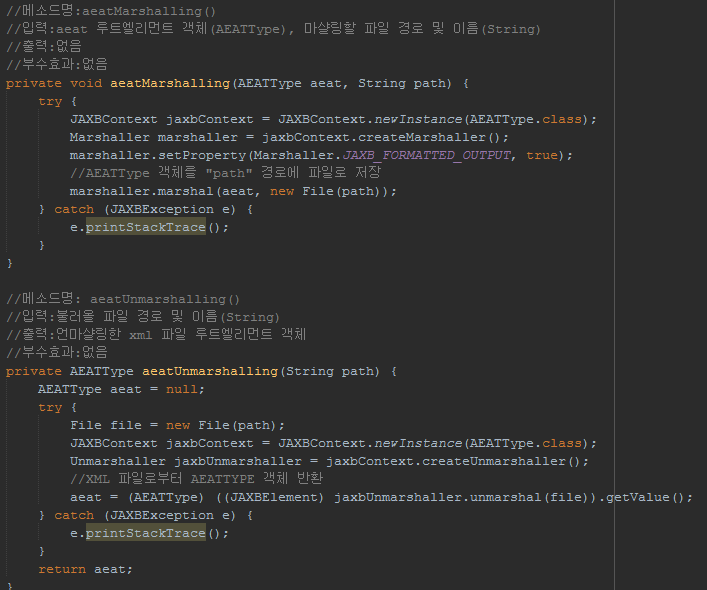


Figure Marshalling / Unmarshalling

구현된 소프트웨어의 Marshalling과 Unmarshalling은 4.3에서 소개한바와 같이 설계되었다. 구현된 Marshalling과 Unmarshalling의 Java code는 위의 Figure 22와 같다.

## JMS Interface

JMS Interface는 ActiveMQ를 통해 구현되었다.

먼저 1:1 방식을 구현하기 위해 ActiveMQ에서 Queue를 만들어서 메시지 송수신을 구현하였다. 1:1방식은 아래 Figure 23과 같이 ActiveMQ에서 Queue를 생성하여 1명의 수신자에게만 메시지를 송신한다.

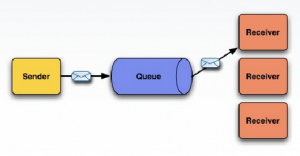


Figure Queue

1:다 방식은 아래 Figure 24와 같이 Topic을 만들어서 다수의 수신자가 메시지를 수신할 수 있도록 한다.

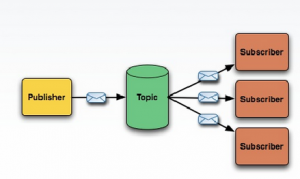


Figure Topic

# Appendix

<http://activemq.apache.org/>

<https://github.com/doorBW/AEATsoftware>

<http://mmlab.uos.ac.kr/Capstone/>

<https://www.atsc.org/candidate-standard/a331-atsc-candidate-standard-signaling-delivery-synchronization-and-error-protection/>

<http://mmlab.uos.ac.kr/Capstone/%EC%8B%A4%EC%8A%B5%EC%9E%90%EB%A3%8C/XMLSchema.pdf>