

소프트웨어실습

요구사항명세서

2017.12.09.

문범우

2013440043

## 차 례

1. 소개(Introduction)	1
1.1. 목표	1
1.2. 범위	1
1.3. 용어 및 정의	1
1.4. 참고문헌	2
1.5. 개요 (Overview)	2
2. 일반사항	3
2.1. 타 시스템과의 관계	3
2.2. 제약사항	10
2.3. 가정	10
2.4. 위험요소	11
3. 요구사항 분석	12
3.1 기능 요구사항(Functional Requirements)	12
3.1.1. 기능요구사항 1	12
3.1.2. 기능요구사항 2	14
3.2 비기능 요구사항 (Nonfunctional Requirements)	15
3.2.1. 시스템 기본요건	15
3.2.2. 기술요구사항	16
3.2.3. 인터페이스요구사항	16
4. 시스템 설계	18
5. 시스템 구현	21
6. Appendix	26

# 초록(Abstract)

본 논문에서는 긴급상황 및 경보 등에서 사용되는 AEAT 를 1:1 방식 또는 1:다 방식으로 메시지를 송수신하기 위해 설계 및 구현된 소프트웨어를 서술한다. 또한 AEAT 형식을 XML schema 로 저장하거나 파일로 저장된 XML schema 를 AEAT 형식으로 불러와 사용자 및 관리자가 보다 빠르고 간편하게 또는 작성 및 수정에 용이하도록 한다.

## 1. 소개(Introduction)

### 1.1. 목표 (Purpose)

- a) AEAT(Advance Emergency Alert Table) 프로그램을 구현한다.
- b) 입력 내용을 Marshalling 하여 XML 파일을 만들어 저장한다.
- c) XML 파일을 UnMarshalling 한다.
- d) ActiveMQ 및 JMS 를 이용하여 메시지를 송수신한다.
- e) Java Swing 에 대하여 학습하고 사용자편의 GUI 를 구현한다.

### 1.2. 범위 (Scope)

- a) 해당 소프트웨어는 AEAT Editor 이다.
- b) 해당 소프트웨어를 통해 특정 사용자에게 긴급한 상황에 대한 메시지를 송신할 수 있다.
- c) 해당 소프트웨어를 통해 특정 사용자는 긴급한 상황에 대한 메시지를 수신할 수 있다.
- d) 프로그램 사용자는 AEAT 양식을 알맞게 작성하여 XML 파일을 생성할 수 있다.
- e) 프로그램 사용자는 ActiveMQ 및 JMS 를 통해 메시지를 송수신 할 수 있다.
- f) 프로그램 사용자는 송수신된 메시지를 AEAT Editor 를 통해 편집할 수 있다.
- g) 프로그램 사용자는 프로그램에서 메시지 송수신 목적지를 설정할 수 있다.
- h) 해당 소프트웨어는 AEAT 양식에 맞추어 필요한 상황을 입력하고, 이를 송수신하거나 XML 파일로 save 및 load 한다.

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

AEAT: Advance Emergency Alert Table

JMS: Java Message Service

## 1.4. 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-Delivery-Sync-FEC-S33-1-788r5.pdf>

## 1.5. 개요 (Overview)

- a) 해당 소프트웨어는 위급상황을 알리기 위한 테이블을 편집하거나 이를 JMS 로 송수신할수 있도록 만들어진 것이다.
- b) 사용자의 편의성을 위해 Radio Button 과 Combo Box 및 Tabbed Pane 을 활용하였다.
- c) 해당 소프트웨어는 AEAT main 에서 프로그램 실행과 버튼에 대한 기능설정을 진행하고, AEAT network 에서 JMS 와 ActiveMQ 를 이용한 메시지 송수신 기능을 수행하며 AEAT view 에서 프로그램의 GUI 를 구성되도록 설계되었다.

## 2. 일반사항 (Overall description)

### 2.1. 타 시스템과의 관계 (Product Perspective)

#### 2.1.1. 시스템 인터페이스(System interface)

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

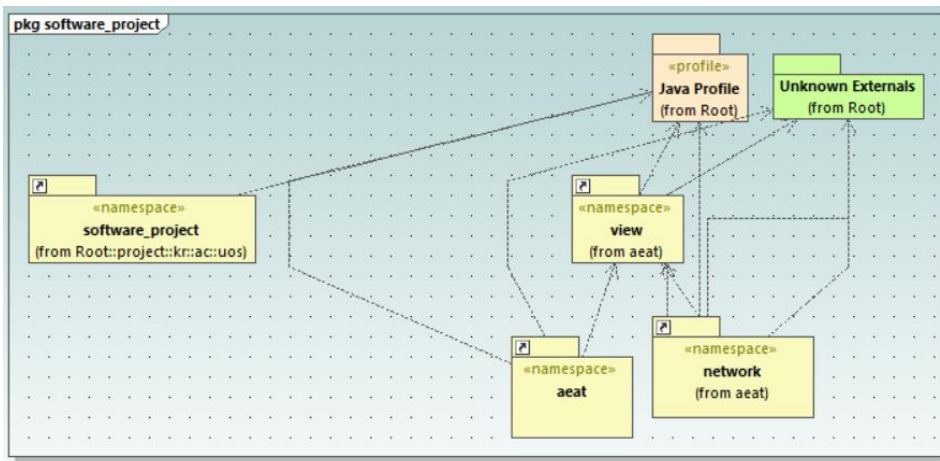


Figure 1 Overall UModel

사용자가 network 를 통해 수신한 메시지는 view 로써 표현되고, 이때 aeat 양식을 따른다. 또한 view 에서 aeat 양식에 따라 작성한 것을 network 를 통해 수신할 수 있다.

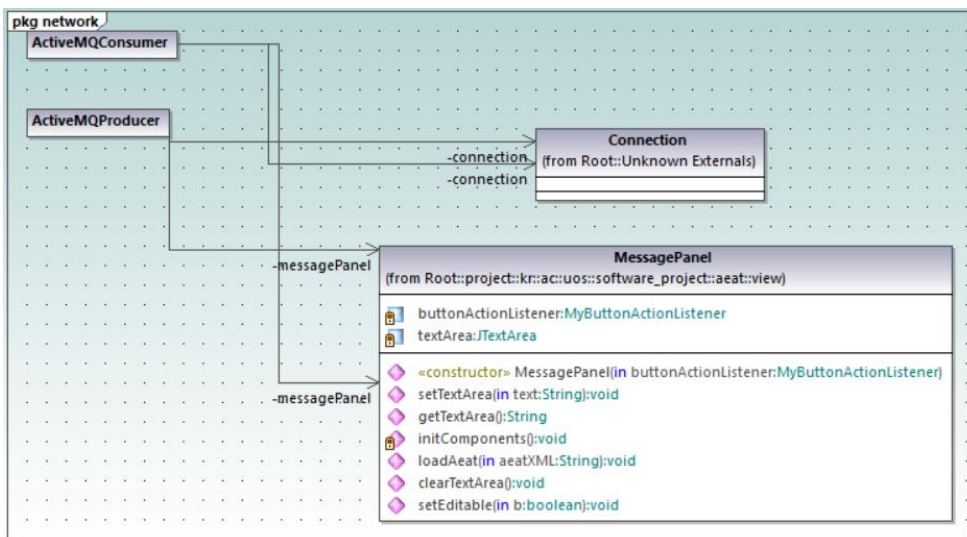


Figure 2 Network UModel

Figure 2 를 통해 Network UModel 을 확인할 수 있다. ActiveMQConsumer 와 ActiveMQProducer 가 함께 Connection 및 MessagePanel 과 연관되어 있다.

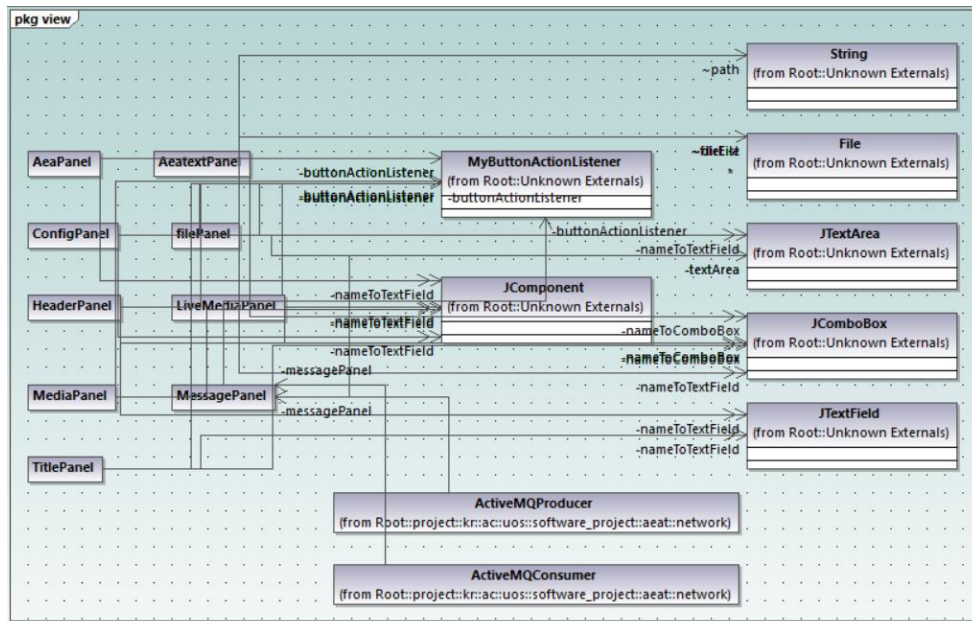


Figure 3 View UModel

Figure 3 을 통해 View 의 UModel 을 확인할 수 있다.

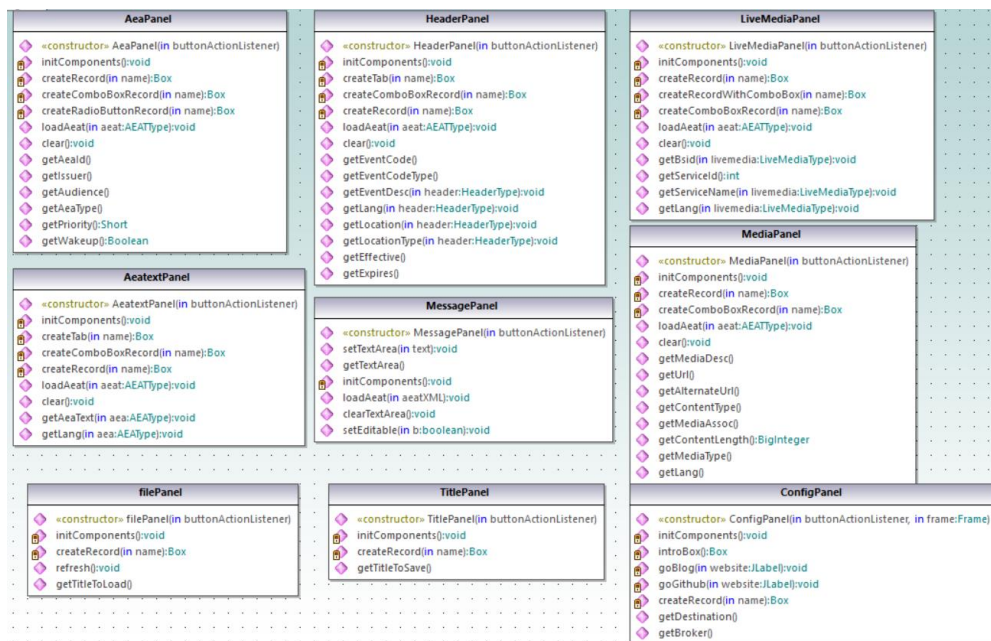


Figure 4 Interface of each Panel

Figure 4 에서 각 패널에 대한 인터페이스를 확인할 수 있다.

AEAT Editor 는 각 Panel 을 통해 구성된다. 각 Panel 에서는 이후 Marshalling 또는 UnMarshalling 을 위한 기능을 수행한다. AEAT Editor 는 view 에서 구현되며, view 는 network 와 연결되어 메시지 송수신이 가능하다.

## 2.1.2. 사용자 인터페이스(User interfaces)

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

**Figure 5 AEAT Editor**

Figure 5 는 AEAT Editor 화면이다. 사용자는 해당 화면에서 xml 파일을 불러와 UnMarshalling 하여 화면의 구성에 맞도록 AEAT 를 확인할 수 있다. 또한 해당 내용을 편집하거나 새로 작성하여 Marshalling 과정을 통해 XML 파일로 저장할 수 있다.

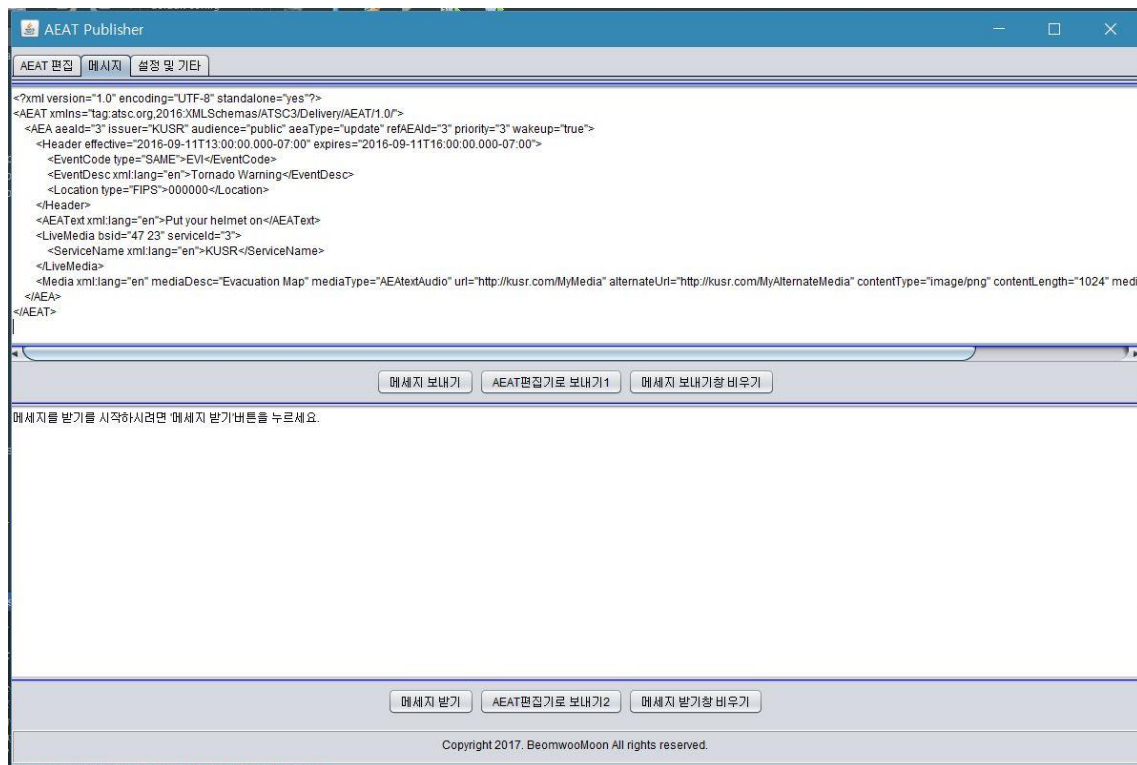
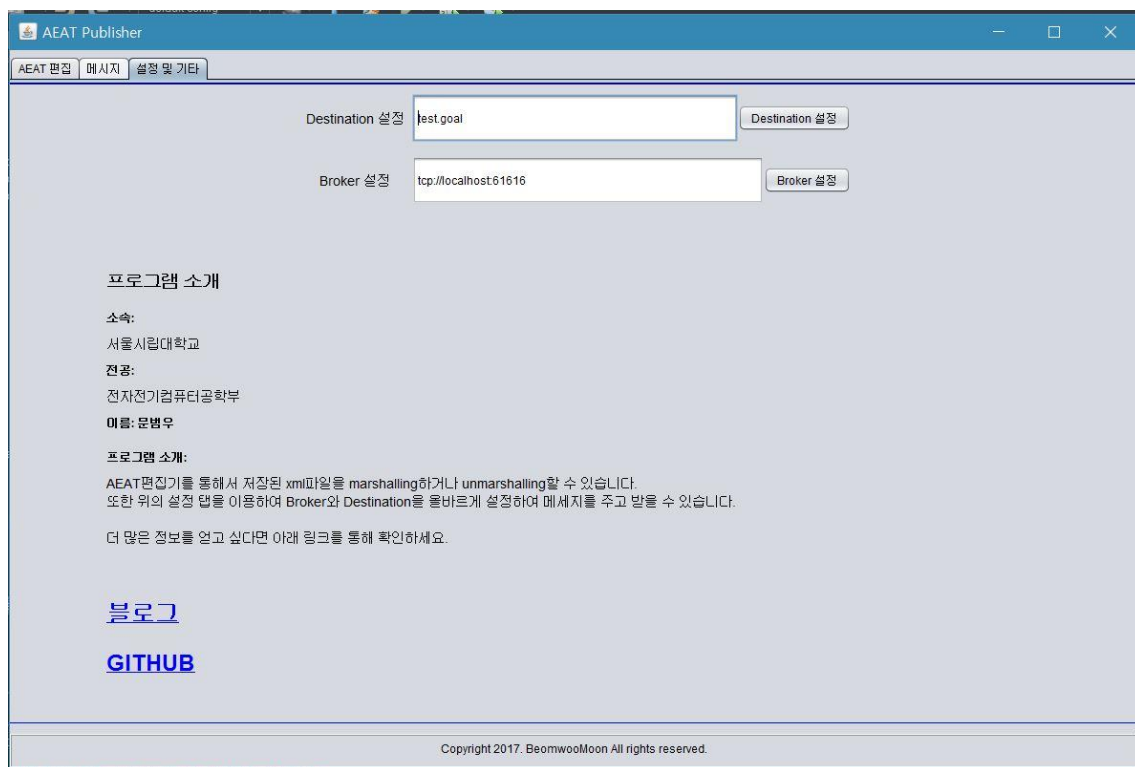


Figure 6 Message View

Figure 6 를 통해 메시지 view 를 확인할 수 있다. 사용자는 앞의 AEAT Editor 를 통해 불러온 XML 파일을 바로 메시지 송신창에 load 되는 것을 확인할 수 있다. 전체적인 view 는 위에서 언급한 바와 같이 network 와 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 Editor 로 전달할 수 있다.

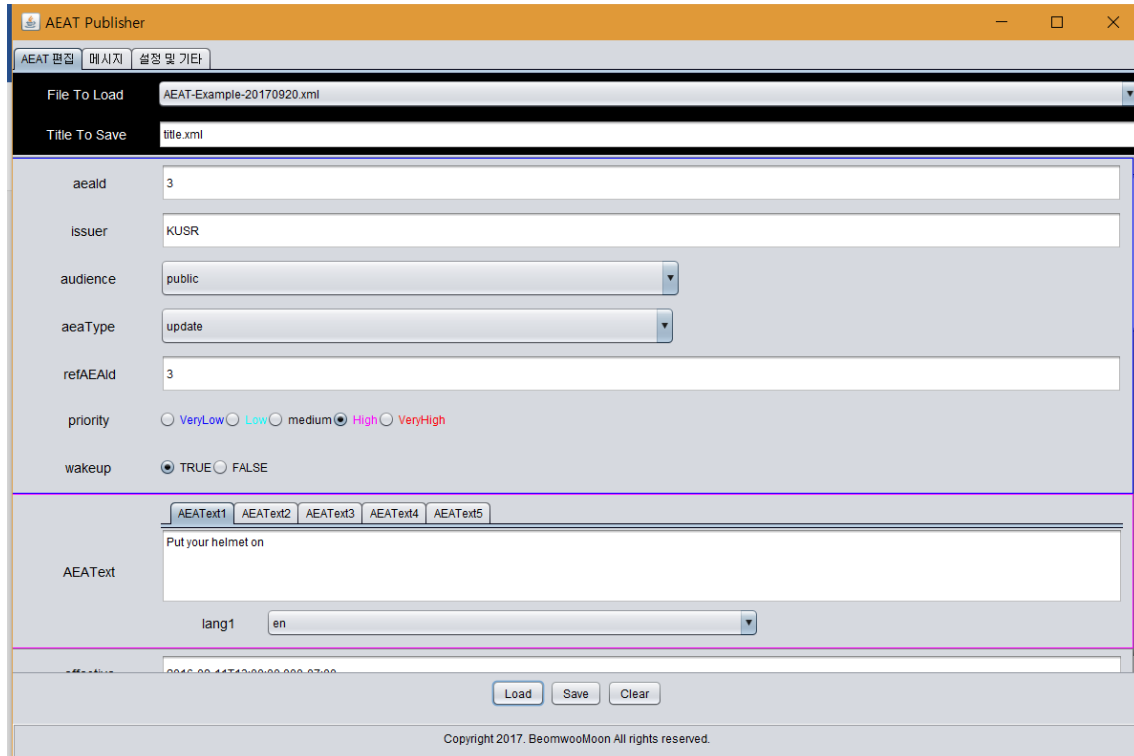


**Figure 7 Config view**

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

### 2.1.3. 운영(Operations)

#### a) Load XML schema



**Figure 8 Load XML schema**

사용자는 Figure 8 에서 볼 수 있는 File To Load 와 Load 버튼을 이용하여 xml 파일로 저장되어 있는 파일을 UnMarshalling 하여 AEAT 양식에 맞춰서 확인할 수 있다.

File To Load 는 “xml/”경로에 있는 파일을 불러온다.

#### b) Save AEAT to XML schema

Figure 8 을 보면 Title To Save 와 Save 버튼을 확인할 수 있다. 사용자는 현재 작성된 AEAT 를 XML schema 로 저장하기 위하여 저장될 파일의 이름을 Title to Save 에서 지정할 수 있으며 Save 버튼을 누르면 해당 이름으로 파일이 저장된다. 파일은 “xml/”경로로 저장되며, 저장이 됨과 동시에 파일목록을 새로 읽어와서 File To Load 에 있는 항목이 업데이트 되기 때문에 새롭게 저장된 xml 파일을 실시간으로 확인할 수 있다.

## c) Send message through ActiveMQ

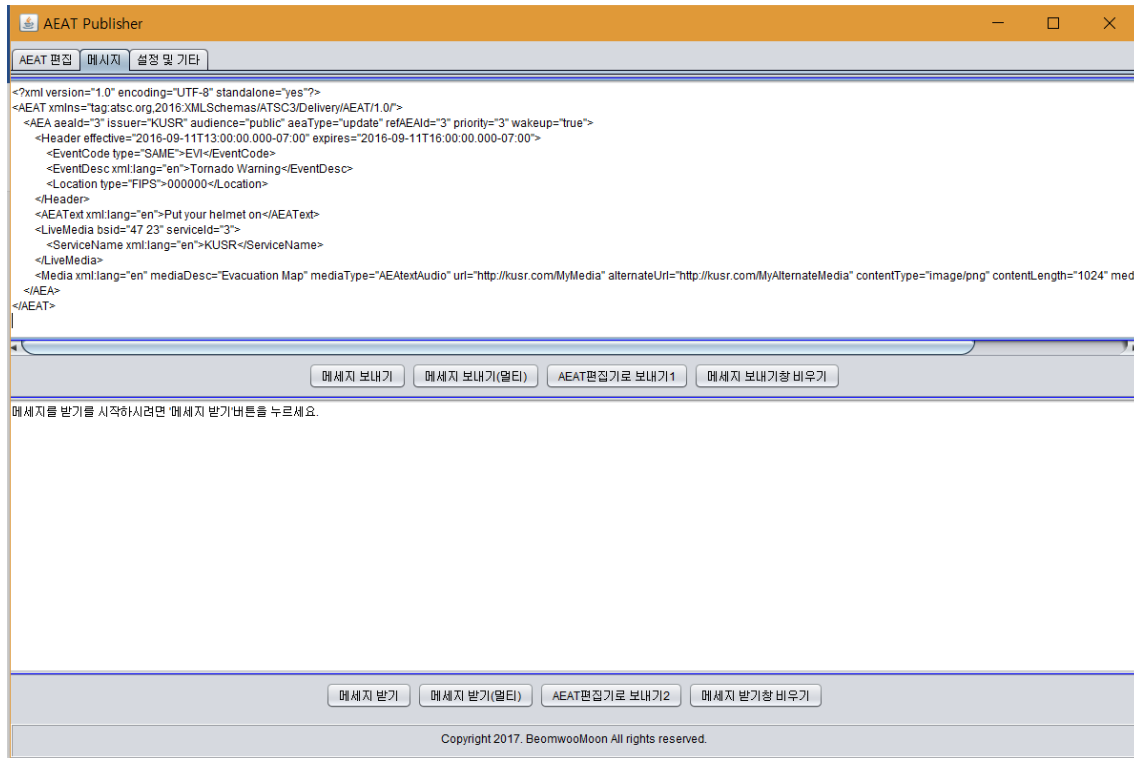


Figure 9 Message view

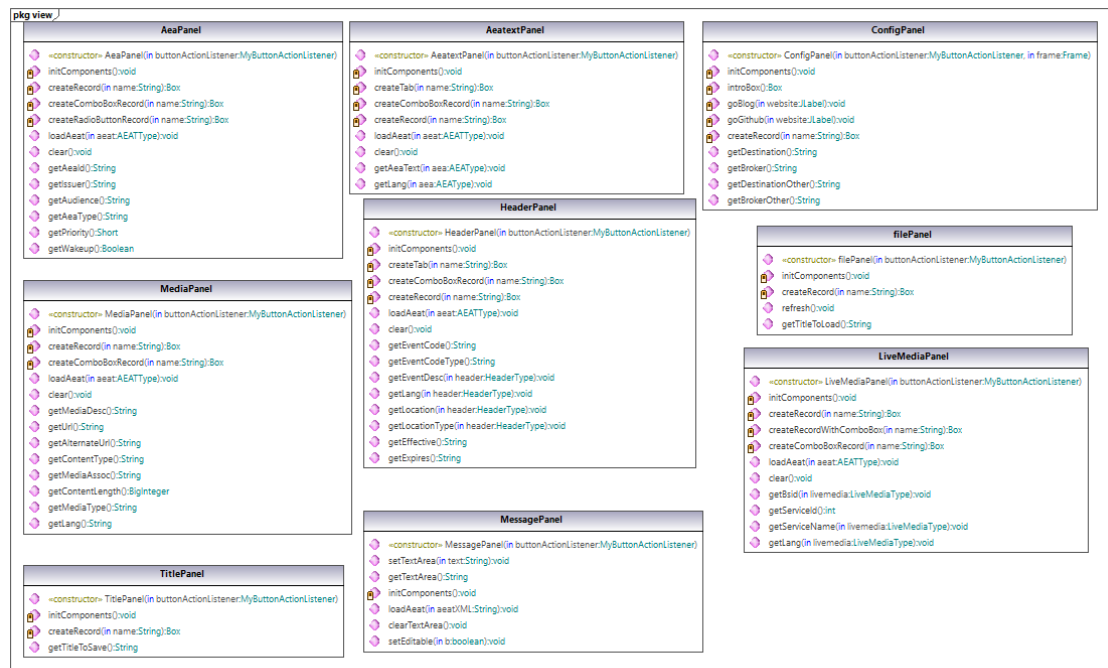
사용자는 위의 Figure 9 에 보이는 Message view 에서 ActiveMQ 를 이용한 메시지 송수신이 가능하다. 1:1 방식으로 메시지를 송수신하기 위해서는 메시지 보내기 및 메시지 받기 버튼을 누르면 되며, 1:다 방식으로 메시지를 송수신하기 위해서는 메시지 보내기(멀티) 및 메시지 받기(멀티)를 누르면 된다.

메시지를 보내면 상단의 입력창의 텍스트가 사라지면서 지정된 Destination 과 Broker 로 메시지가 송신된다.

## d) Receive message through ActiveMQ

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

## 2.2. 제약사항 (Product functions)



Generated by UModel

www.altova.com

**Figure 10 Function**

Figure 10 을 통해 전체적인 Function 을 확인할 수 있다.

- 사용자는 직접 AEAT 를 작성하여 XML schema 로 저장할 수 있다.
- 사용자는 XML schema 파일을 불러와 AEAT 양식에 맞게 확인할 수 있다.
- 사용자는 AEAT 를 다른 사용자에게 송신할 수 있다.
- 사용자는 AEAT 를 다른 사용자로부터 수신할 수 있다.

## 2.3. 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 재난 관련 메시지 및 기상 메시지 또는 기타 긴급성이 필요로 되는 메시지를 XML schema 로 송신이 가능하다. 직접적으로 XML schema 에 맞춰서 메시지를 작성하지않고 간단하게 해당 소프트웨어의 AEAT 편집기를 이용하여 필요한 메시지를 작성할 수 있으며 이를 XML schema 로 저장하여 추후 필요한 경우 확인할 수 있다.

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

## 2.4. 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.

## 2.5. 가정 및 종속성 (Assumptions and dependencies)

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

### 3. 요구사항 분석

#### 3.1. 기능 요구사항 (Functional Requirements)

해당 소프트웨어는 기능적으로 AEAT 구현과 ActiveMQ 를 이용한 메시지 송수신을 요구한다. 추가적으로 AEAT 를 사용자가 작성 및 편집하기 용이하도록 RadioButton 과 ComboBox 및 TabbedPane 의 적절한 사용을 요구한다.

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

##### 3.1.1.기능요구사항 1: AEAT 구현

본 절에서는 AEAT 를 구현하기 위해 AEAT 의 전체적인 구조를 살펴본다. 아래 트리는 XMLspy 를 통해서 도출하였다.

a) AEAT Type

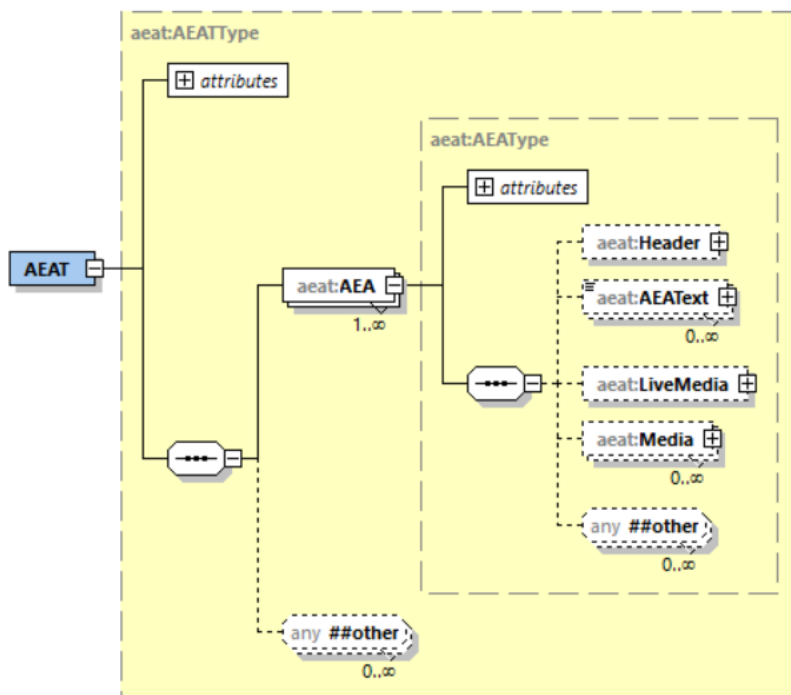


Figure 11 Original AEAT Type schema

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

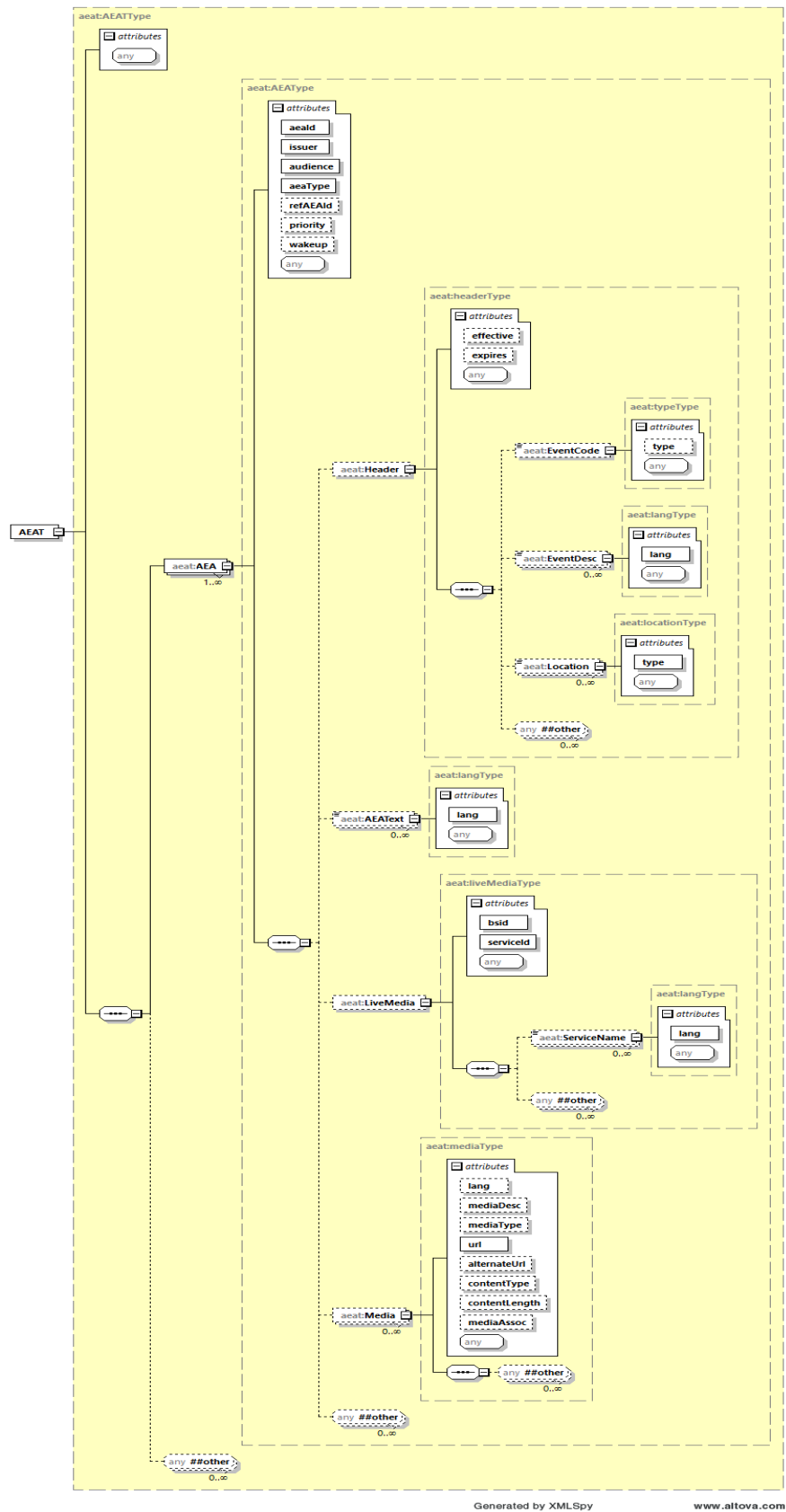
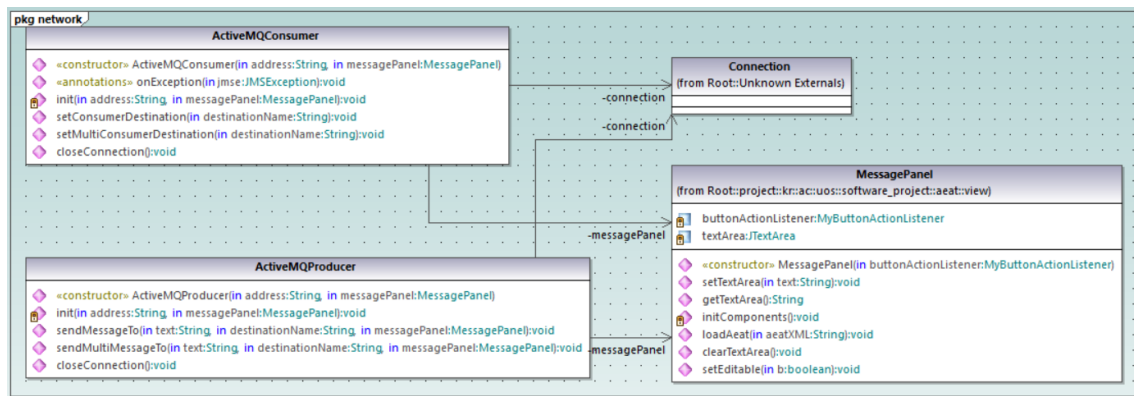


Figure 12 Overall AEAT Type

### 3.1.2.기능요구사항 2: 메세지 송수신

메시지 송수신을 구현하기 위해 ActiveMQ 를 활용한다.

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



**Figure 13 Overall of Message Function**

ActiveMQConsumer 와 ActiveMQProducer 가 함께 Connection 과 MessagePanel 과 연관되어 있다. 메시지를 송신하기 위해 ActiveMQProducer 가 Connection 을 통해 지정된 서버와 연결이 된다. 이후 MessagePanel 에 입력된 text 를 송신한다. 비슷하게, 메시지를 수신하기 위해 ActiveMQConsumer 가 Connection 을 통해 지정된 서버와 연결이 된다. 이후 MessagePanel 에 수신된 text 를 출력한다.



## 3.2. 비기능 요구사항 (Nonfunctional Requirements)

### 3.2.1. 시스템 기본요건

#### 3.2.1.1. 성능

해당 소프트웨어는 사용자가 작성 및 수정한 AEAT 가 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.

#### 3.2.1.2. 신뢰성

사용자 및 관리자가 해당 소프트웨어를 통해 AEAT 형식을 XML schema 로 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 data 에 있어서 손상이 없도록 한다.

#### 3.2.1.3. 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.

#### 3.2.1.4. 복구성

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

#### 3.2.1.5. 사용자인터페이스

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

## 3.2.2. 기술요구사항

### 3.2.2.1. 소프트웨어

- 해당 소프트웨어를 설계 및 개발하기 위해 Java Programming 능력을 필요로 한다.
- AEAT 형식을 다루기 위해 Marshalling 및 UnMarshalling 에 대한 기초 개념을 필요로 한다.
- 메시지 송수신을 구현하기 위해 ActiveMQ 에 대한 사용법을 필요로 한다.

## 3.2.3. 인터페이스요구사항

### 3.2.3.1. 인터페이스 기능

- AEAT 를 편집할 수 있는 인터페이스가 존재하여 사용자 및 관리자가 간편하게 AEAT 를 편집한다.
- 수정 및 보완된 AEAT 가 XML schema 로 저장될 수 있어 관리 및 유지보수가 용이하도록 한다.
- 과거의 XML schema 를 불러와 AEAT 형식으로 볼 수 있는 인터페이스가 존재한다.
- 사용자가 작성한 AEAT 또는 불러온 AEAT 를 제 3 자에게 송신할 수 있다.
- 제 3 자로부터 AEAT 형식 및 text 를 수신할 수 있다.
- 제 3 자로부터 수신된 AEAT 형식의 XML schema 를 AEAT 편집기로 전달할 수 있다.

### 3.2.3.2. 전달하는 데이터 유형 및 포맷

- 사용자 및 관리자가 ActiveMQ 를 통해 송신하는 메시지 형식은 크게 XML schema 와 text 이다. 그 중 XML schema 는 아래 Figure 14 와 같다.

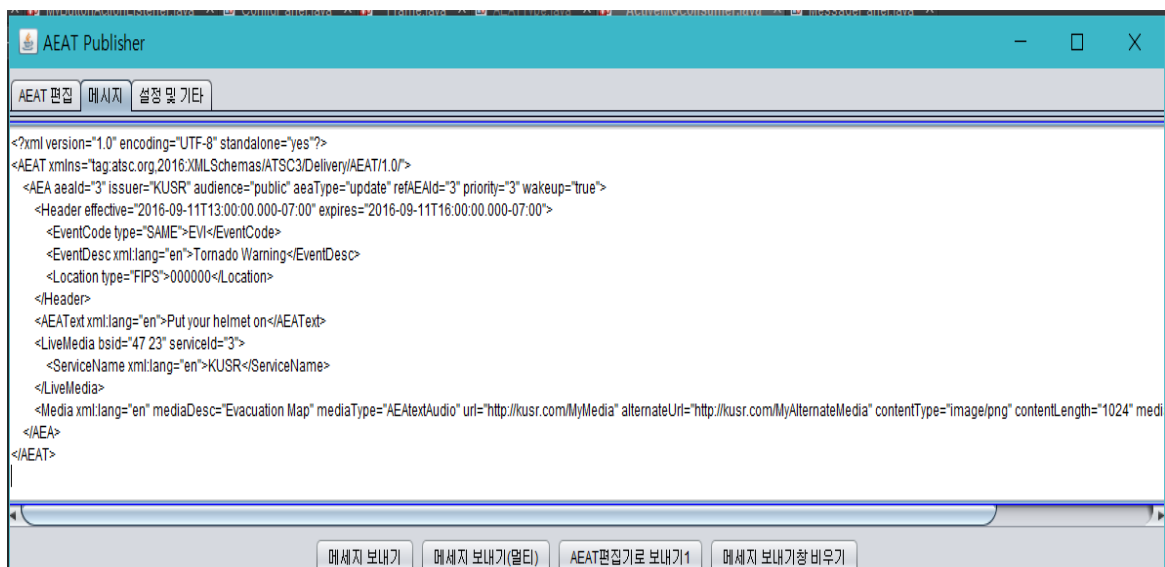


Figure 14 Format of sending message

### 3.2.3.3. 전달받는 데이터 유형 및 포맷

- 사용자 및 관리자가 ActiveMQ 를 통해 수신하는 메시지 형식은 크게 XML schema 와 text 이다. 그 중 XML schema 는 아래 Figure 15 와 같다



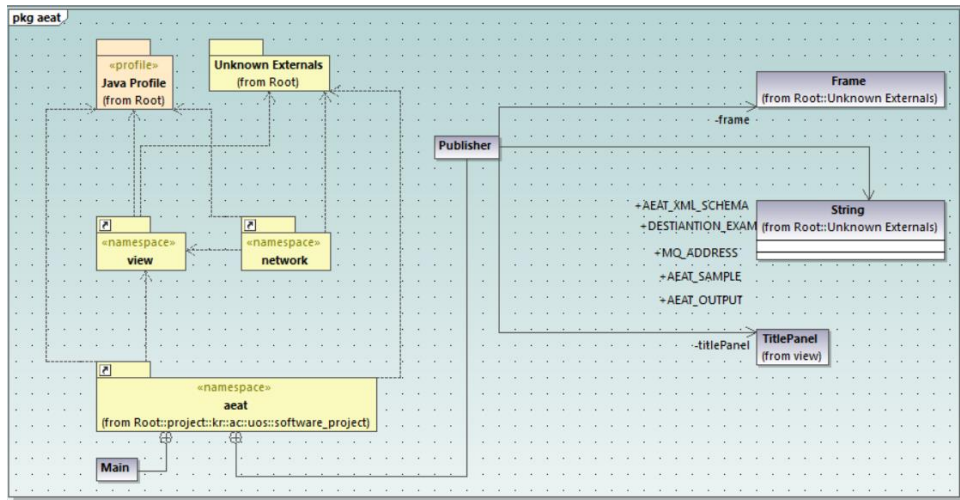
Figure 15 Format of receiving message

## 4. 시스템 설계

### 4.1. GUI

- AEAT 편집기, 메시지, 설정 및 기타의 항목을 가진 총 3 개의 TabbedPane 을 구현한다.
- AEAT 편집기 내부에서는, AEAT 형식에 알맞게 작성할 수 있도록 항목을 구성한다.
- AEAT 편집기 내부에서 특정 값을 갖는 항목에 대해서는 RadioButton 또는 ComboBox 를 이용한다.
- AEAT 편집기 내부에서 하나 이상의 값을 갖는 항목에 대해서는 TabbedPane 을 이용한다.
- AEAT 편집기에서는 작성된 AEAT 를 XML schema 로 저장될 수 있도록 파일 이름을 작성하는 항목과 저장 버튼이 구현된다.
- AEAT 편집기에서는 XML schema 를 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 Destination 과 Broker 를 설정할 수 있도록 구현된다.
- 설정 및 기타 창에서는 프로그램에 대한 간략한 소개와 개발자 소개가 포함된다.

## 4.2. MODEL

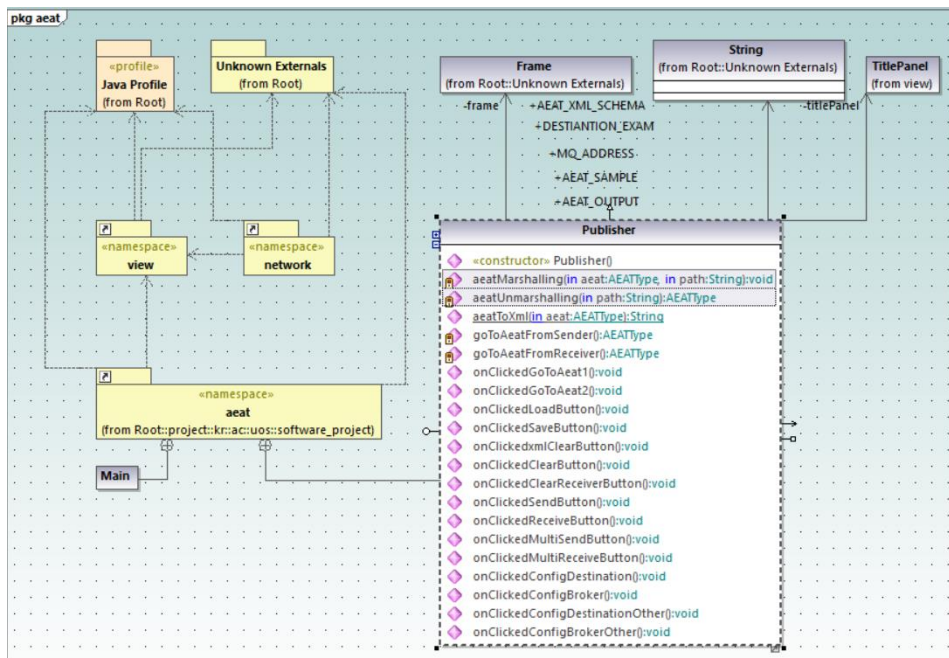


**Figure 16 Overall MODEL**

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

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

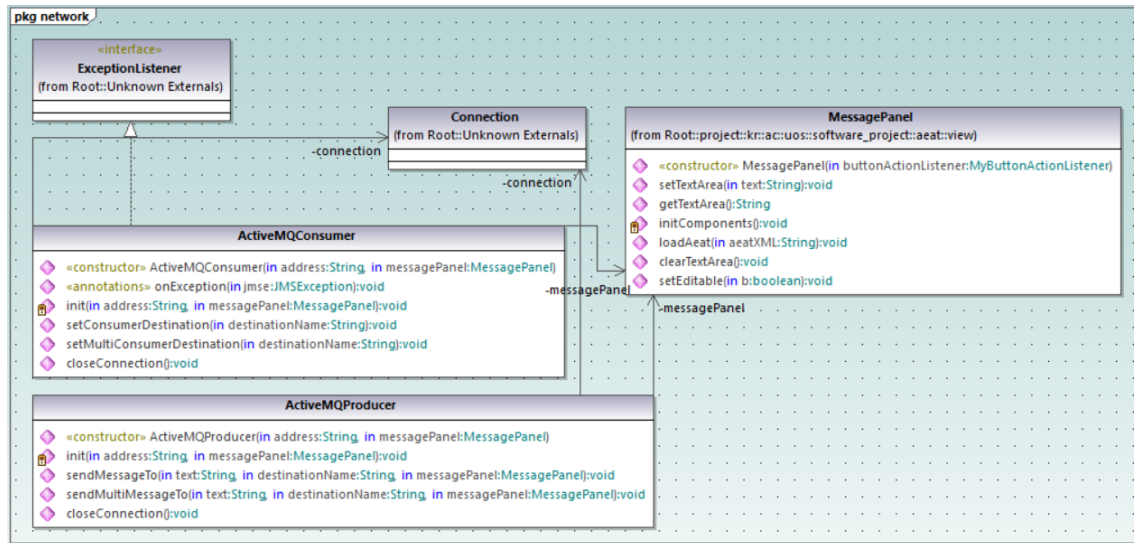
## 4.3. Marshalling/Unmarshalling



**Figure 17 Marshalling/Unmarshalling**

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

#### 4.4. JMS Interface



**Figure 18 JMS Interface through ActiveMQ**

Figure 18 에서 확인할 수 있듯이 JMS 는 ActiveMQ 를 통해 구현된다. 메시지를 송신하기 위해 ActiveMQProducer 에서 Connection 이 이루어지며 sendMessageTo 를 통해 1:1 방식의 메시지 송신이 가능하며 sendMultiMessageTo 를 통해 1:다 방식의 메시지 송신이 가능하다. 또한 ActiveMQConsumer 에서 Connection 이 이루어지며 setConsumerDestination 을 통해 1:1 방식의 메시지 수신이 이루어지며, setMultiConsumerDestination 을 통해 1:다 방식의 메시지 수신이 이루어진다.

## 5. 시스템 구현

### 5.1. GUI

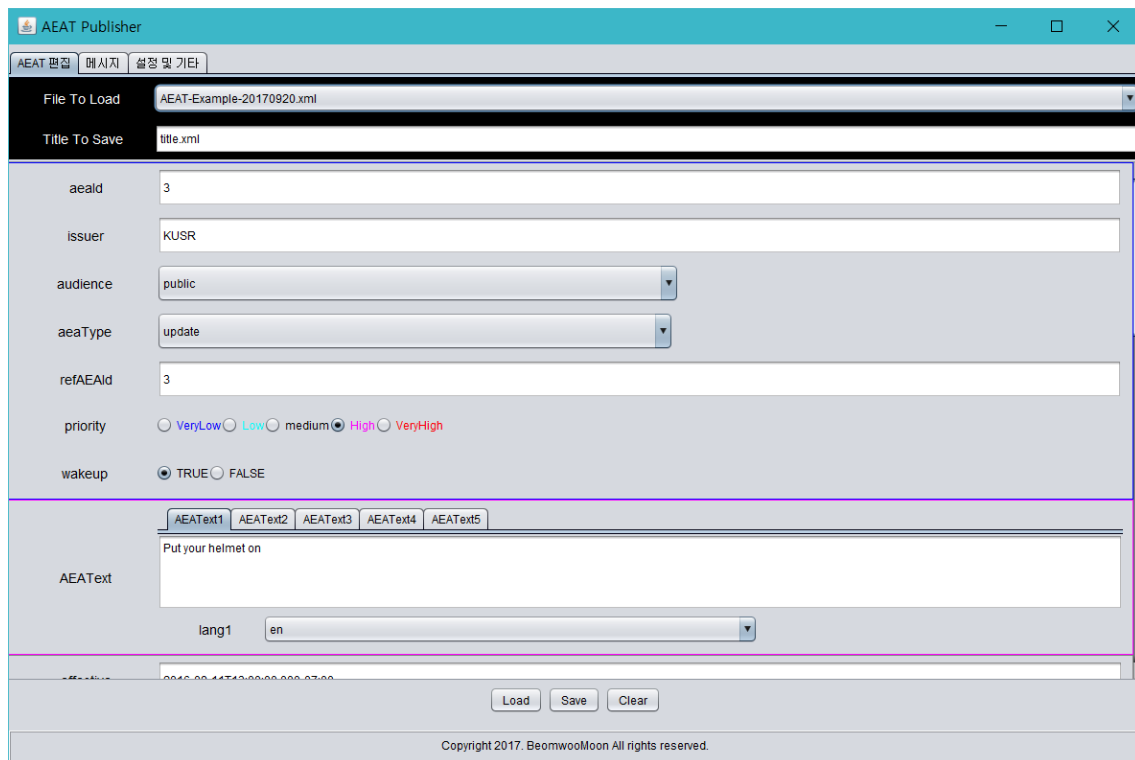
아래 Figure 19,20,21 을 통해 구현된 소프트웨어의 GUI 를 확인할 수 있다.

Figure 19 를 확인하면, 상단에 File To Load 와 Title To Save 를 통해 AEAT 를 XML schema 로 저장하거나 저장된 XML schema 를 AEAT 로 불러올 수 있다.

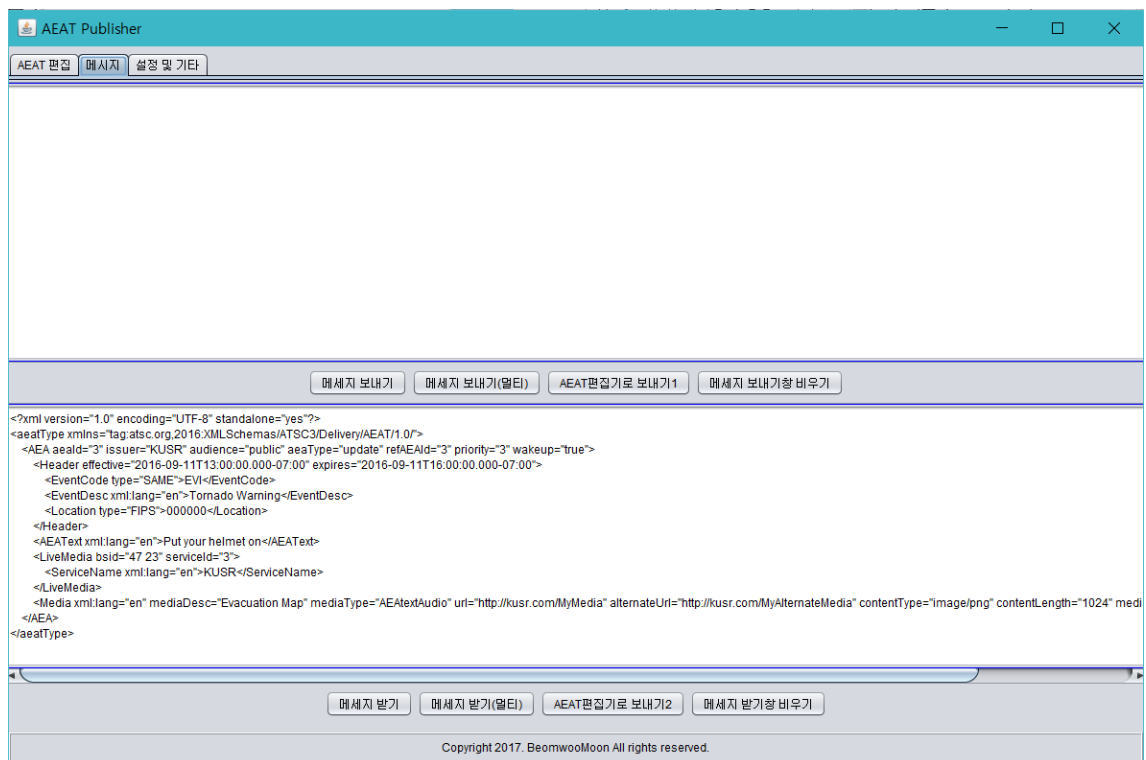
또한 각 항목에 대해서 작성 및 수정이 용이하도록 항목이 구성되었으며 audience 및 aeaType 등 특정 값으로 정해져 있는 항목은 ComboBox 를 활용하거나 priority 와 같이 RadioButton 을 이용하였다. 그리고 AEAText 와 같이 여러 개의 내용을 포함할 수 있는 항목은 그 내부에서 TabbedPane 을 가지도록 구현하였다.

Figure 20 을 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 메시지 송수신을 사용자가 선택할 수 있다. 또한 각 창에 있는 내용을 Figure 19 의 AEAT 편집기로 전달할 수 있도록 버튼이 구현되어 있다. 추가적으로 창의 내용을 비우기 위한 버튼이 구현되어 있다.

Figure 21 을 확인하면 Destination 과 Broker 를 설정할 수 있는 설정창이 구현되었다. 메시지 메 올바르게 송수신하기 위해 2 개의 Destination 과 2 개의 Broker 를 설정할 수 있도록 구현하였고 이를 설정하면 console 창에 설정이 적용되었다는 출력이 표시된다. 또한 하단에는 간단한 프로그램 소개와 개발자에 대한 소개가 나와 있다.



**Figure 19 AEAT Editor**



**Figure 20 Message send or receive**



The screenshot shows the 'Config' tab of the 'AEAT Publisher' application. The interface has a teal header bar with the title 'AEAT Publisher' and standard window controls. Below the header is a tab bar with 'AEAT 편집', '메시지', and '설정 및 기타', with '설정 및 기타' being the active tab. The main content area is light gray and contains four configuration sections, each with a text input field and a '설정' (Set) button:

- Destination 설정**: Input field contains 'djjproject'.
- Broker 설정**: Input field contains 'tcp://djjproject.com:61616'.
- 상대 Destination**: Input field contains 'djjproject'.
- 상대 Broker**: Input field contains 'tcp://djjproject.com:61616'.

Below these sections is a '프로그램 소개' (Program Introduction) section. It lists the author as '문범우' (Moon Beom-woo) from '서울시립대학교 전자전기컴퓨터공학부' (Department of Electronic and Electrical Computer Engineering, Seoul National University of Education). It describes the program's function: using AEAT editor to marshal/unmarshal XML files and send/receive messages via Broker and Destination. It includes links to a '블로그' (Blog) and 'GITHUB'.

At the bottom of the window, a footer bar contains the text: 'Copyright 2017. BeomwooMoon All rights reserved.'

Figure 21 Config tab

## 5.2. MODEL

구현된 소프트웨어의 MODEL 은 4.2 에서 소개한바와 같이 설계되었다.

View 와 network, aeat 가 서로 연관되어 있으며 이들이 유기적으로 데이터를 교환한다.

## 5.3. Marshalling/Unmarshalling

```
//메소드명:aeatMarshalling()
//입력:aeat 루트엘리먼트 객체(AEATType), 마샬링할 파일 경로 및 이름(String)
//출력:없음
//부수효과:없음
private void aeatMarshalling(AEATType aeat, String path) {
    try {
        JAXBContext jaxbContext = JAXBContext.newInstance(AEATType.class);
        Marshaller marshaller = jaxbContext.createMarshaller();
        marshaller.setProperty(Marshaller.JAXB_FORMATTED_OUTPUT, true);
        //AEATType 객체를 "path" 경로에 파일로 저장
        marshaller.marshal(aeat, new File(path));
    } catch (JAXBException e) {
        e.printStackTrace();
    }
}

//메소드명: aeatUnmarshalling()
//입력:불러올 파일 경로 및 이름(String)
//출력:언마샬링한 xml 파일 루트엘리먼트 객체
//부수효과:없음
private AEATType aeatUnmarshalling(String path) {
    AEATType aeat = null;
    try {
        File file = new File(path);
        JAXBContext jaxbContext = JAXBContext.newInstance(AEATType.class);
        Unmarshaller jaxbUnmarshaller = jaxbContext.createUnmarshaller();
        //XML 파일로부터 AEATTYPE 객체 반환
        aeat = (AEATType) ((JAXBElement) jaxbUnmarshaller.unmarshal(file)).getValue();
    } catch (JAXBException e) {
        e.printStackTrace();
    }
    return aeat;
}
```

**Figure 22 Marshalling / Unmarshalling**

구현된 소프트웨어의 Marshalling 과 Unmarshalling 은 4.3 에서 소개한바와 같이 설계되었다.

구현된 Marshalling 과 Unmarshalling 의 Java code 는 위의 Figure 22 와 같다.

## 5.4. JMS Interface

JMS Interface 는 ActiveMQ 를 통해 구현되었다.

먼저 1:1 방식을 구현하기 위해 ActiveMQ 에서 Queue 를 만들어서 메시지 송수신을 구현하였다. 1:1 방식은 아래 Figure 23 과 같이 ActiveMQ 에서 Queue 를 생성하여 1 명의 수신자에게만 메시지를 송신한다.

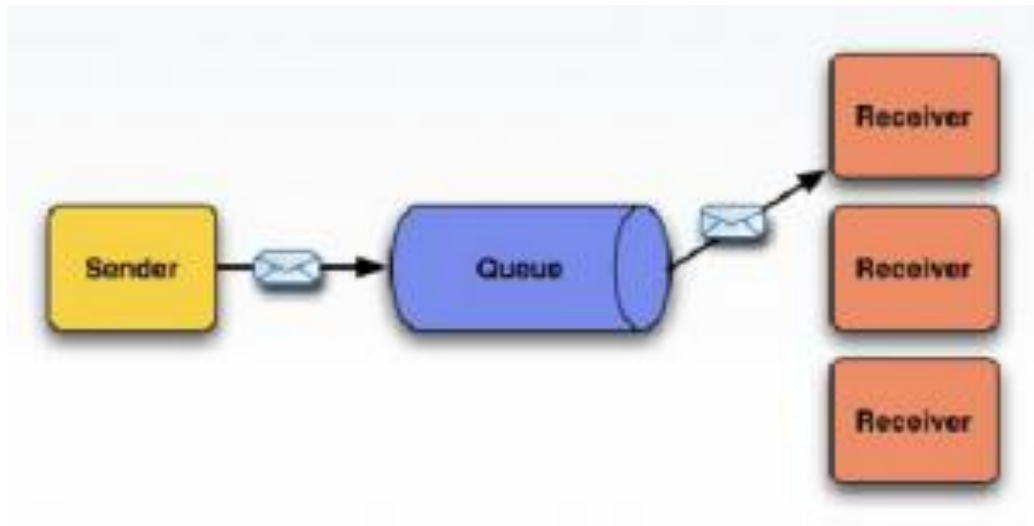


Figure 23 Queue

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

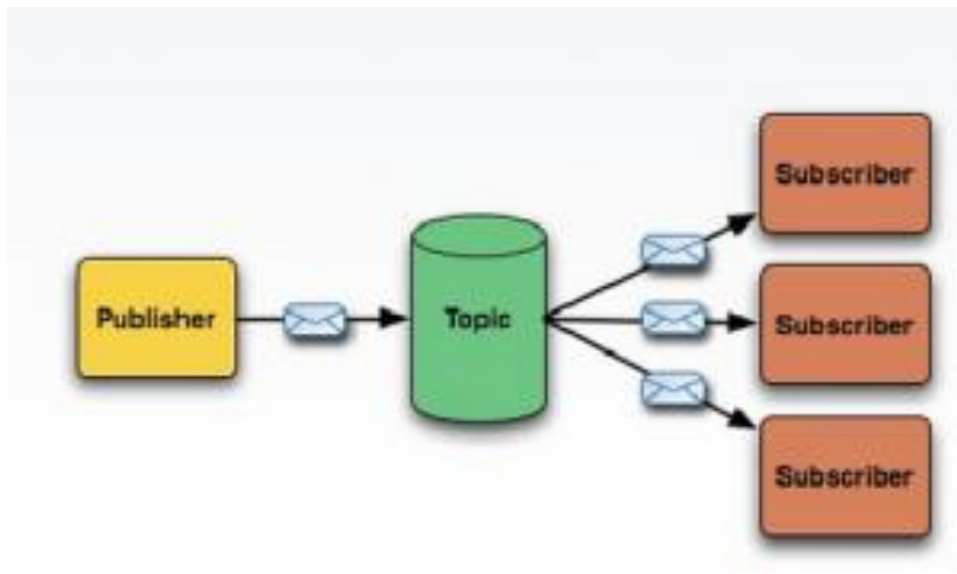


Figure 24 Topic

## 6. 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>