

MPEG-IoMT 기반 미디어 사물 자동 연결 및 통신을 위한 미디어 사물 관리자 제안

김신, 박병준, 이예지, 윤경로
건국대학교
new.xin22@gmail.com

Proposal of Media Thing Manager based on MPEG-IoMT for Automatic Media Thing Connection and Communication

Shin Kim, Byeong Jun Park, Yegi Lee, Kyoungro Yoon
Konkuk University

요 약

MPEG-IoMT 표준은 미디어 사물과 관련된 데이터 포맷 및 인터페이스와 관련된 국제 표준 프로젝트로서 여러 플랫폼에서도 다양한 미디어 사물을 등록, 연결 및 통신을 할 수 있게 제공한다. 하지만 현재로서 다수의 미디어 사물 간 연결 수립은 사용자가 수동으로 수행해야 하며, 연결이 되었다 하더라도 미디어 사물 간 데이터 통신의 진행 상태를 확인할 수단이 없다. 따라서 본 논문에서는 MPEG-IoMT 표준 내 미디어 사물 관리자(Media Thing Manager)를 제안한다. 미디어 사물 관리자는 미디어 사물 간 자동 연결을 위해 미션 데이터를 생성 및 관리하며, API 를 통해 미디어 사물 간 통신 수행 여부를 확인할 수 있다.

1. 서론

MPEG-IoMT 표준은 동영상, 영상, 오디오 등 미디어를 다룰 수 있는 미디어 사물과 관련된 데이터 포맷, 인터페이스 등을 제공하는 표준이다. 카메라, 오디오와 같은 미디어 사물은 미디어 데이터 자체로 사용자에게 정보를 제공할 수 있지만, 미디어 사물을 분석하여 데이터를 만드는 분석기와 같은 경우는 다른 미디어 사물에서 얻어온 미디어 정보를 필요로 한다. 즉, 미디어 사물의 다양한 기능을 사용하기 위해서 다른 미디어 사물과의 연결 및 통신이 요구된다.

MPEG-IoMT 표준은 미디어 사물 간 자동 연결에 대한 데이터 포맷과 API 등이 부재하며, 미디어 사물 간 연결을 위해서는 사용자가 직접 discovery API 를 이용하여 연결해야

한다[1]. 이는 번거로운 작업이며, 복잡한 미디어 사물 기반 시스템을 구축할 경우 어려움이 있을 수 있다. 또한 미디어 사물 간 연결이 되어있더라도 미디어 사물의 기능 수행 상태를 파악하기 어렵다는 단점이 있다. 이러한 문제점을 해결하기 위해 다수의 미디어 사물 간 자동 연결 및 통신 상태를 확인할 수 있는 관리자가 필요하다.

본 논문에서는 미디어 사물 간 자동 연결 및 통신을 관리할 수 있는 미디어 사물 관리자에 대하여 제안한다. 미디어 사물 관리자는 이용하고자 하는 미디어 사물의 정보를 취합하여 미션 데이터를 생성하며, 미션 데이터를 이용하여 미디어 사물이 미션을 이해하여 타 미디어 사물과 통신을 자동으로 수립하며, 미디어 사물 간 통신 상태를 확인할 수 있다.

본 논문의 구성은 다음과 같다. 2 절에서는 MPEG-IoMT 내 미디어 사물 간 자동 연결을 위한 미션 데이터에 대해 제안한다.

다음으로 3 절에서는 2 절에서 제안한 미션 데이터를 기반으로 한 미디어 사물 관리자의 프로토타입에 대해 제안하며 마지막으로 4 절에서는 미디어 사물 관리자의 추후 적용 가능성과 관련하여 본 논문의 마무리를 맺는다.

2. 미션 데이터

미션은 미디어 사물이 해야할 일을 의미한다. 즉, 각 미디어 사물은 다수의 미디어 사물들과 연결되어 있을 때, 특정 미디어 사물에게 어떤 데이터를 요청하고 어떠한 결과물을 받아야 하는지 알아야 한다. 그림 1 은 IoMT Camera, IoMT Video Content Class Generator, IoMT Storage 간 데이터 및 API 의 흐름도이다.

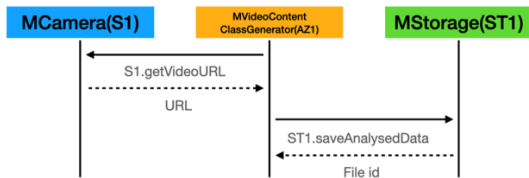


그림 1. IoMT Camera, Video Content Class Generator, Storage 간 데이터 및 API 흐름도[2]

IoMT Video Content Class Generator 는 동영상의 클래스를 분석하는 분석기로서, IoMT Camera 로부터 동영상을 취득해야 한다. 따라서 getVideoURL 함수를 이용하여 IoMT Camera 로부터 동영상의 URL 을 얻어오며, 해당 동영상을 분석하여 분석 데이터를 생성하고 이를 저장하기 위해 IoMT Storage 에 saveAnalysedData 를 분석한 데이터인 analysedData 을 파라미터로서 요청하며 저장에 성공한 경우 file id 를 반환한다.

미션 데이터는 미디어 사물 간 연결 및 통신이 수월하게 진행될 수 있도록 기타 정보를 제공한다. 즉, 미션 데이터는 그림 1 과 같이 다른 미디어 사물 간의 연결 및 통신을 위한 정보를 포함해야 하며, 추가적으로 기능 수행 상태까지 확인할 수 있어야 한다.

그림 2 는 제안하는 미션 데이터의 예제다. 각 미션 데이터는 미션 데이터의 id 정보로 구분할 수 있으며, 미션에는 MThing(사용하는 미디어 사물), request(입력 파라미터), output(입력 파라미터와 capability 에 따른 출력 정보), fulfillmentState(미션 수행 상태) 정보가 포함되어 있다. 각 미션은 순차적으로 진행되며 미션이 수행되면 해당 미디어 사물은 fulfillmentState 의 값을 변경한다.

```

<missionData id='MD-001'>
  <mission>
    <MThing id='MS001'>MCamera</MThing>
    <request>None</request>
    <Output>URL</Output>
    <fulfillmentState>False</fulfillmentState>
  </mission>
  <mission>
    <MThing id='MA001'>
      MVideoContentClassGenerator</MThing>
    <request>URL</request>
    <Output>AnalysedData</Output>
    <fulfillmentState>False</fulfillmentState>
  </mission>
  <mission>
    <MThing id='MST001'>MStorage</MThing>
    <request>AnalysedData</request>
    <Output>FileID</Output>
    <fulfillmentState>False</fulfillmentState>
  </mission>
</missionData>
    
```

그림 2. 그림 1 기반 미션 데이터 예제

3. 미디어 사물 관리자 프로토타입

미디어 사물 자체가 특정 다른 미디어 사물을 찾고 연결하는 것은 상당히 복잡한 시스템을 요구한다. 따라서 다수의 미디어 사물들을 연결하고 미디어 사물 끼리 데이터 통신을 하기 위해서는 미션을 관리할 수 있는 중간 매체가 필요하다. 미디어 사물 관리자는 여러 미디어 사물 간 연결 및 통신을 관리하는 미디어 사물으로써, 앞 절에서 제안한 미션 데이터를 기반으로 미디어 사물 간 연결 및 통신 상태에 대해 확인할 수 있어야 한다.

미디어 사물 관리자는 이용하고자 하는 미디어 사물 간 미션 데이터를 생성하기 위해 앞서 미디어 사물들의 정보를 취합해야 한다. 표 1 은 그림 1 기반 미디어 사물들의 정보를 나타낸다.

표 1. 그림 1 기반 미디어 사물 정보(미디어 사물 오름차순)

MThing	MThing Type	Capability	Input Parameter	Output Data
Camera	MSensor	CAPTURE_VIDEO	None	URL
Storage	MStorage	STORAGE_SAVE	AnalysedData	File ID
VCCG	MAnalyser	ANALYSE_VIDEO_CONTENT_CLASS	URL	AnalysedData

표 2. 그림 X 기반 미디어 사물 정보(미디어 사물 관리자에 의해 재정렬)

MThing	MThing Type	Capability	Input Parameter	Output Data
Camera	MSensor	CAPTURE_VIDEO	None	URL
VCCG	MAnalyser	ANALYSE_VIDEO_CONTENT_CLASS	URL	AnalysedData
Storage	MStorage	STORAGE_SAVE	AnalysedData	File ID

표 2은 미디어 사물에 의해 재정렬된 미디어 사물 정보로, 먼젓번 미디어 사물의 출력 결과물과 해당 미디어 사물의 입력 파라미터가 동일하게 테이블을 재정렬한다.

```

Before re-ordering
[[{'mvideocontentclassgenerator', 'MAB02', '1.2', ['ANALYSE_VIDEO_CONTENT_CLASS', ('url', 'analysed data')]}]
[{'mstorage', 'MST001', '1.3', ['STORAGE_SAVE', ('analysed data', 'fileid')]}]
[{'mcamera', 'MS001', '1.1', ['SENSOR_CAPTURE_VIDEO', (0, 'url')]}]

After re-ordering
[{'mcamera', 'MS001', '1.1', ['SENSOR_CAPTURE_VIDEO', (0, 'url')]}]
[[{'mvideocontentclassgenerator', 'MAB02', '1.2', ['ANALYSE_VIDEO_CONTENT_CLASS', ('url', 'analysed data')]}]
[{'mstorage', 'MST001', '1.3', ['STORAGE_SAVE', ('analysed data', 'fileid')]}]

Generated Mission Data
<?xml version="1.0"?>
<missionData id="12">
  <mission>
    <MThingInfo id="MS001">mcamera</MThingInfo>
    <requestNone/>
    <outputurl/>
    <fulfillmentState>false</fulfillmentState>
  </mission>
  <mission>
    <MThingInfo id="MAB02">mvideocontentclassgenerator</MThingInfo>
    <requesturl/>
    <output>analysed data</output>
    <fulfillmentState>false</fulfillmentState>
  </mission>
  <mission>
    <MThingInfo id="MST001">mstorage</MThingInfo>
    <request>analysed data</request>
    <output>fileid</output>
    <fulfillmentState>false</fulfillmentState>
  </mission>
</missionData>
    
```

그림 3. 미션 데이터 정보 재정렬 및 생성 미션 데이터 예제

그림 3은 표 1, 2의 수행 결과 및 미디어 사물 정보를 이용하여 미션 데이터를 충분히 생성할 수 있음을 보여준다.

그림 4는 미디어 사물 관리자의 미션 데이터 초기화, 설정 및 수행 상태 확인 프로세스를 나타낸다. 미디어 사물 관리자는 모든 미디어 사물에 미션 데이터를 초기화하기 위해 setMissionData 함수를 통해 미션 데이터를 브로드캐스트하며, 미디어 사물 관리자는 미션 데이터가 수행되고 있는 중간 getFulfillmentState 함수를 통해 어디까지 진행되었는지 확인할 수 있다. 그림 5, 6은 그림 4기반 미디어 사물 관리자의 미션 데이터 설정 및 수행 상태 확인 여부를 확인할 수 있는 데모를 보여준다.

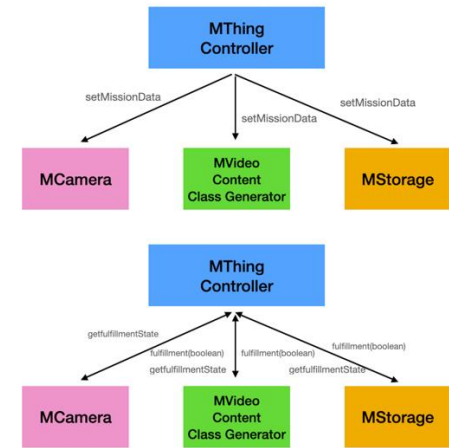


그림 4. 미디어 사물 관리자의 미션 데이터 및 통신 여부 확인 예상 흐름도

```

connection succeeded
Received Mission Data : <?xml version="1.0"?>
  <missionData id="12">
    <mission>
      <MThingInfo id="MS001">mcamera</MThingInfo>
      <requestNone/>
      <outputurl/>
      <fulfillmentState>false</fulfillmentState>
    </mission>
    <mission>
      <MThingInfo id="MAB02">mvideocontentclassgenerator</MThingInfo>
      <requesturl/>
      <output>analysed data</output>
      <fulfillmentState>false</fulfillmentState>
    </mission>
    <mission>
      <MThingInfo id="MST001">mstorage</MThingInfo>
      <request>analysed data</request>
      <output>fileid</output>
      <fulfillmentState>false</fulfillmentState>
    </mission>
  </missionData>
  This MThing's Request : none, Output : url

connection succeeded
Received Mission Data : <?xml version="1.0"?>
  <missionData id="12">
    <mission>
      <MThingInfo id="MS001">mcamera</MThingInfo>
      <requestNone/>
      <outputurl/>
      <fulfillmentState>false</fulfillmentState>
    </mission>
    <mission>
      <MThingInfo id="MAB02">mvideocontentclassgenerator</MThingInfo>
      <requesturl/>
      <output>analysed data</output>
      <fulfillmentState>false</fulfillmentState>
    </mission>
    <mission>
      <MThingInfo id="MST001">mstorage</MThingInfo>
      <request>analysed data</request>
      <output>fileid</output>
      <fulfillmentState>false</fulfillmentState>
    </mission>
  </missionData>
  This MThing's Request : url, Output : analysed data

connection succeeded
Received Mission Data : <?xml version="1.0"?>
  <missionData id="12">
    <mission>
      <MThingInfo id="MS001">mcamera</MThingInfo>
      <requestNone/>
      <outputurl/>
      <fulfillmentState>false</fulfillmentState>
    </mission>
    <mission>
      <MThingInfo id="MAB02">mvideocontentclassgenerator</MThingInfo>
      <requesturl/>
      <output>analysed data</output>
      <fulfillmentState>false</fulfillmentState>
    </mission>
    <mission>
      <MThingInfo id="MST001">mstorage</MThingInfo>
      <request>analysed data</request>
      <output>fileid</output>
      <fulfillmentState>false</fulfillmentState>
    </mission>
  </missionData>
  This MThing's Request : analysed data, Output : fileid
    
```

그림 5. 미션 데이터 생성 프로토 타입 결과물 (setMission)

```

<MThingInfo id="MS001">mcamera</MThingInfo>
<requestNone/>
<outputurl/>
<fulfillmentState>false</fulfillmentState>
</mission>
<mission>
  <MThingInfo id="MAB02">mvideocontentclassgenerator</MThingInfo>
  <requesturl/>
  <output>analysed data</output>
  <fulfillmentState>false</fulfillmentState>
</mission>
<mission>
  <MThingInfo id="MST001">mstorage</MThingInfo>
  <request>analysed data</request>
  <output>fileid</output>
  <fulfillmentState>false</fulfillmentState>
</mission>
</missionData>
This MThing's Request : none, Output : url
Requesting getFulfillmentState
connection succeeded
Requesting getFulfillmentState
connection succeeded

Received FulfillmentState : false
Requesting getFulfillmentState
connection succeeded
Requesting getFulfillmentState
connection succeeded

Received FulfillmentState : false
Requesting getFulfillmentState : True
connection succeeded
Requesting getFulfillmentState : True
connection succeeded

Received FulfillmentState : false
Requesting getFulfillmentState : False
connection succeeded
Requesting getFulfillmentState
connection succeeded
1. setMission
2. getFulfillment
    
```

그림 6. 미션 데이터 생성 프로토 타입 결과물 (getFulfillmentState)

4. 결론

본 논문에서는 MPEG-IoMT 를 위한 미션 데이터 포맷 정의 및 미디어 사물 관리자 프로토타입에 대하여 제안하였다. 미디어 사물 관리자는 선택된 미디어 사물들의 정보를 취합하고 재정렬하여 미션 데이터를 생성하고, 만들어진 미션 데이터를 통해 미디어 사물 간 연결을 자동으로 수립하였으며 통신 및 기능 수행 상태를 확인할 수 있었다.

하지만 현재 미디어 사물 관리자 프로토타입은 복잡한 시나리오에 대한 미션 데이터를 고려하지 않은 상태이므로 향후 실제 미디어 사물 시스템 구축을 위해 프로토타입 확장이 필요할 것으로 예상된다.

감사의 글

이 논문은 2021 년 정부(산업통상자원부)의 재원으로 한국산업기술평가관리원의 지원을 받아 수행된 연구임. (20002212, 미디어 사물 인터넷 거래 및 컴팩트 데이터 표현을 위한 표준 기술 개발)

참고문헌

- [1] N00028, "Text of ISO/IEC DIS 23093-2 IoMT Discovery and Communication API", 2020
- [2] m56887, Shin Kim, Yegi Lee and Kyoungro Yoon, "Study of mission data handling with MThing Controller in MPEG-IoMT", 2021