

MPEG-IoMT 내 MissionDiagram을 위한 Finite Machine 조사 및 MissionDiagram 서술 도구 제안

김신*, 이예지, 윤경로

건국대학교

*new.xin22@gmail.com

Development of Content and Running Environment based on SCORM-ACAP Interoperability

Shin Kim*, Yegi Lee, Kyoungro Yoon

Konkuk University

요약

MPEG-IoMT(Internet of Media Thing)은 국제 표준으로서, 미디어 사물 간 데이터를 주고 받을 수 있도록 데이터 포맷과 API를 정의하고 있고 있기 때문에 본 표준을 통해 호환되지 않는 모든 미디어 사물 간에도 쉽게 연결될 수 있을 것이다. 하지만, 현재 여러 미디어 사물 간 자동화된 연결에 대한 표준은 존재하지 않아 미디어 사물을 연결하고자 할 때는 사용자가 표준 API를 통해 직접 연결을 해야 하는 단점이 있어 미디어 사물 간 자동 연결이 가능하게 특정 기능에 대한 필요한 미디어 사물을 제공할 수 있는 서술 도구에 대한 필요성이 나타나고 있다. 따라서, 본 논문에서는 Finite Machine 기반으로 미디어 사물 간 자동화된 연결을 제공할 수 있는 Mission Diagram 서술 도구에 대해 제안한다.

1. 서론

MPEG-IoMT(ISO/IEC 23093)은 미디어를 다룰 수 있는 사물 간 표준화된 데이터 포맷, API 등을 제공하는 국제 표준이다. 미디어 사물은 오디오 혹은 비디오를 다룰 수 있는 카메라, 마이크론 등 뿐만 아니라 오디오, 비디오를 분석할 수 있는 컴퓨터 노드 등 여러 종류로 나눌 수 있다. 카메라, 마이크론과 같이 센서링 노드의 경우 MSensor로, 소스를 분석할 수 있는 분석 노드의 경우 MAnalyzer, 정보를 저장하는 노드의 경우 MStorage, 어떠한 행동을 작동시키는 노드를 MActuator라 지칭한다.

MPEG-IoMT 국제 표준을 통해 미디어 사물 간 연결 및 통신이 가능하나, 현재로서 미디어 사물 간 자동 연결에 대한 부분에 대한 표준은 부재하다. 그래서, 사용자가 여러 미디어 사물을 연결하여 기능을 이용하고 싶을 경우 사용자가 직접 표준 API를 사용하여 기능 별로 미디어 사물을 검색하고 사용 가능 여부를

확인하여 연결하여야 한다.

본 논문에서는 Finite Machine에 대한 리뷰를 바탕으로 MPEG-IoMT 를 위한 Mission Diagram을 서술하는 도구를 제안한다. 다양한 미디어 사물 간 자동화된 연결을 위해 기능에 따른 미디어 사물의 Mission 을 파악하고, 이를 Diagram으로 제공하여 자동화된 연결이 가능하도록 하고자 한다. 본 논문의 구성은 다음과 같다. 2장에서 Mission Diagram을 위해 필요한 배경 지식으로 Finite Machine에 대해 조사하고, 3장에서는 배경 지식을 바탕으로 Mission Diagram 서술 도구를 제안한다. 마지막으로 4장에서 본 논문에 대한 결론을 맺는다.

2. Finite-state Machine

Finite Machine은 Finite-state Machine 또는 Finite-state Automata를 줄여서 표현한 것으로, 컴퓨터 프로그램과 전자 논리 회로를 설계할 때 쓰이는 수학적 모델을 말한

다. Finite-stateMachine은 유한한 개수의 상태를 가질 수 있는 오토마타라 할 수 있으며, 이러한 기계는 한 번에 오로지 하나의 상태만을 가지게 된다. 현재 상태란 임의의 주어진 시간의 상태를 말하며, 이러한 기계는 어떠한 사건에 의해 한 상태에서 다른 상태로 변화할 수 있고 이를 전이라고 한다. 특정 유한 오토마타는 현재 상태에서부터 가능한 전이 상태와, 이러한 전이를 유발하는 조건들의 집합으로서 정의될 수 있다. Finite-state Machine 종류에는 Mealy Machine, Moore Machine 등이 있다.

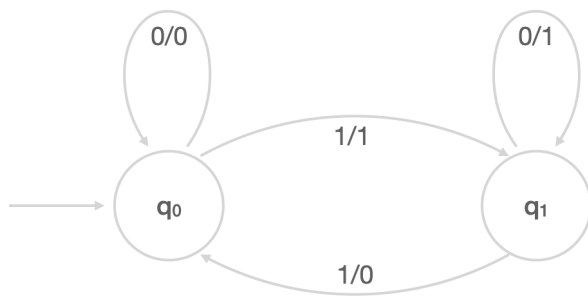


그림 1. Mealy Machine 그래프 예제

Mealy Machine[1]은 현재 상태와 전이에 사용되는 입력에 의해 출력이 정해지는 Finite-state Machine이다. Mealy Machine은 상태들의 유한 집합, 초기 상태, 입력 알파벳의 유한 집합, 출력 알파벳의 유한 집합, 전이 함수, 출력 함수 등 6개의 원소 쌍으로 구성되어 있다. 그림 1은 Mealy Machine의 한 예를 나타내는 그래프이다. 원형 노드는 현재 상태를 말하며, 화살표는 전이를 의미한다. Mealy Machine은 화살표에 그림 1과 같이 입력과 출력을 모두 표현할 수 있으며, 슬래시 왼쪽은 입력, 오른쪽은 출력을 의미한다.



그림 2. Moore Machine 그래프 예제

그에 비해 Moore Machine[2]은 현재 상태에 따라 출력이 정해지는 Finite-state Machine이다. 즉, 어느 상태에 놓을 때마다 연관된 심볼이 출력되는 것을 의미한다. Moore Machine도 Mealy Machine과 동일하게 상태들의 유한 집합, 초기 상태,

입력 알파벳의 유한 집합, 출력 알파벳의 유한 집합, 전이 함수, 출력 함수 등 6개의 원소 쌍으로 구성되어 있다. 그림 2는 Moore Machine 예를 보여주는 그래프로서, Mealy Machine과 상동하게 원형 노드는 현재 상태, 화살표는 전이를 의미한다. 단지, 원형 노드 안에 현재 상태뿐만 아니라 해당 상태로 전이할 시 출력이 포함되어 있다.

Mealy Machine과 Moore Machine은 출력을 만들어 내는 방식에서 차이가 있지만, 같은 문제에 대해 다른 방식으로 문제를 해결한다고 할 때 두 Finite-state Machine은 동치성을 가질 수 있다. 그림 1과 그림 2는 각 Machine에 대한 그래프 예제이지만, 동일한 Finite Machine에 대한 그래프를 표현하고 있다.

3. Mission Diagram 서술 도구 제안

MPEG-IoMT 표준을 지원하는 특정 미디어 사물을 이용하고 할 때, 해당 미디어 사물과 타 미디어 사물간의 연결, 통신 등 해야 할 일을 Mission이라고 정의한다. 따라서 타 미디어 사물과의 소통을 쉽게 파악하기 위한 Diagram을 Mission Diagram이라 개설할 수 있다. Mission Diagram을 바탕으로 추후 미디어 사물 간 간편한 연결 및 통신이 가능하도록 Mission Diagram을 서술할 수 있는 도구가 필요하게 된다.

따라서 본 논문에서 Finite Machine 중 Mealy Machine 형태 기반으로 MPEG-IoMT 내 Mission Diagram을 표현하기 위한 서술 도구를 제안하고자 한다. Mealy Machine의 경우, 출력이 입력과 현재 상태에 의해 결정되기 때문에, 상태에 따른 입출력 표현이 Moore Machine보다 좀 더 간결하게 표현될 수 있다.

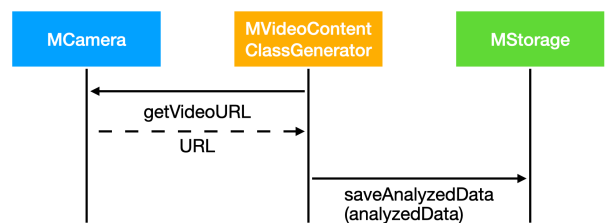


그림 3. MPEG-IoMT 내 미디어 사물 간 통신 예시

그림 3은 MCamera (MSensor), MVideoContent ClassGenerator (MAnalyzer), MStorage 간 간단한 API 및 데이터 통신을 나타내는 흐름도이다. MVideoContent ClassGenerator가 분석을 위해 MCamera에 getVideoURL API를 호출하면, 비디오의 URL을 반환한다. MVideoContent ClassGenerator가 MCamera로부터 받은 URL을 기반으로 해당 비디오의 클래스를 분석하고, 분석한 데이터를 MStorage에 저장하기 위해 saveAnalyzedData를 호출하고, 분석된 데이터

는 저장된다. 즉, 그림 3을 통해 MVideoContent ClassGenerator의 비디오 클래스 분석 기능을 이용하고 싶을 때, 비디오 소스에 해당하는 MCamera와 이를 저장하기 위한 MStorage가 필요하다는 것을 알 수 있다. Mission Diagram은 특정 미디어 사물의 기능을 사용하고 싶을 때 필요한 다른 미디어 사물과 그에 맞는 통신 과정을 나타낼 수 있다. 그림 4는 그림 3에 대한 Mission Diagram으로 만든 예시이다.

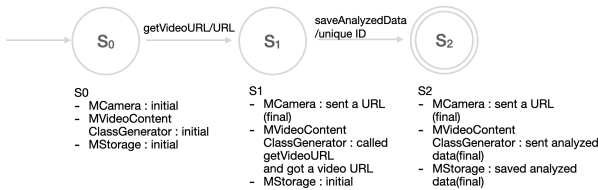


그림 4. 그림 3에 대한 Mission Diagram 예시

그림 4와 같이 Mission Diagram 내 상태는 여러 미디어 사물에 대한 상태를 내포하고 표현할 수 있다. 즉, 미디어 사물은 자신의 상태를 Mission Diagram을 통해 알 수 있고, 필요한 미디어 사물과 주고 받아야할 API 및 데이터에 대해 알 수 있다.

그림 4에는 총 3가지의 상태가 포함되어 있으며, 각 상태에 MCamera, MVideoContentClassGenerator, MStorage의 세부 상태에 대해 서술하고 있다. S0의 경우 세 가지 미디어 사물 모두 초기 상태임을 의미한다. 이후 첫 번째 상태 전이시 getVideoURL의 입력을 통해 URL이라는 출력을 보이게 되는데, 이는 Mealy Machine과 달리 입력의 경우 API를, 출력의 경우 해당 API의 반환 값을 의미한다. 다시 말해, S0의 초기 상태에서 MCamera 내 getVideoURL API가 호출된다면, URL을 반환하면서 S1로 상태가 전이됨을 뜻한다. URL이 반환되면, S1로 상태 전이가 되며 해당 URL은 MVideoContent ClassGenerator가 취득하게 된 상태이며, MCamera의 경우 종료 상태가 된다. 취득한 URL 소스를 기반으로 분석을 한 후 분석한 데이터를 저장하고자 할 때 분석한 데이터를 저장하는 함수인 saveAnalyzedData가 호출되고 저장이 성공하면 저장된 파일의 String 형태의 unique ID를 얻기 되며 S2로 상태 전이가 된다. S2의 상태일 때, MVideoContentClassGenerator와 MStorage는 종료 상태가 되며 최종적으로 Mission Diagram도 종료 상태가 된다.

그림 5는 그림 4 기반 Mission Diagram을 XML 표현한 예시이다. 그림 5와 같이 Mission Diagram을 XML로 표현하여 블록 체인 상에 올린다면, 사용자가 일일이 미디어 사물을 탐색하지 않아도 Mission Diagram기반 XML 데이터를 통해 필요한 미디어 사물 간 자동 연결 및 데이터 통신이 가능해질 것으로 예상된다.

```

<state id = "S0">
  <MStorage>
    <StateDescription/>Initial
  </MStorage>
  <MCamera>
    <StateDescription/>Initial
  </MCamera>
  <MVideoContentClassGenerator>
    <StateDescription/>Initial
  </MVideoContentClassGenerator>
  <MStorage>
    <StateDescription/>Initial
  </MStorage>
  <InputAPI>getVideoURL</InputAPI>
  <OutputData>URL</OutputData>
</state>
<state id = "S1">
  <MStorage>
    <StateDescription/>Sent a URL (final)
  </MStorage>
  <MCamera>
    <StateDescription/>Sent a URL (final)
  </MCamera>
  <MVideoContentClassGenerator>
    <StateDescription/>Called getVideoURL and got a video URL
  </MVideoContentClassGenerator>
  <MStorage>
    <StateDescription/>Initial
  </MStorage>
  <InputAPI>saveAnalyzedData</InputAPI>
  <OutputData>String</OutputData>
</state>
<state id = "S2">
  <MStorage>
    <StateDescription/>Sent an analyzed data (final)
  </MStorage>
  <MCamera>
    <StateDescription/>Sent an analyzed data (final)
  </MCamera>
  <MVideoContentClassGenerator>
    <StateDescription/>Saved an analyzed data (final)
  </MVideoContentClassGenerator>
  <MStorage>
    <StateDescription/>Saved an analyzed data (final)
  </MStorage>
  <InputAPI>None</InputAPI>
  <OutputData>None</OutputData>
</state>
  
```

그림 5. 그림 4에 대한 XML 표현 예시

4. 결론

본 논문에서는 Finite Machine 에 대해 조사하고, Mealy Machine 포맷을 활용하여 MPEG-IoMT 내 Mission Diagram을 위한 서술 도구를 제안하였다. Mealy Machine 포맷을 활용하여 여러 미디어 사물 간 Mission Diagram의 예시를 제작하였으며, 또한 Mission Diagram을 바탕으로 쉽게 읽을 수 있는 XML으로 표현하였다.

그러나, 많은 미디어 사물에 대한 Mission Diagram 표현 가능 여부를 판단해야 할 것이며, 추가적으로 Mission Diagram의 XML 기반 표준 서술 도구에 대한 방향성에 대해 토론이 필요할 것으로 예상된다.

감사의 글

이 논문은 2020년 정부(산업통상자원부)의 재원으로 한국 산업기술개발사업의 지원을 받아 수행된 연구임. (20002212, 미디어 사물 인터넷 거래 및 컴팩트 데이터 표현을 위한 표준 기술 개발)

참 조 문 헌

[1] Mealy, George H, "A Method for Synthesizing Sequential Circuits.", Bell System Technical Journal. pp. 1045-1079, 1955

[2] Moore E. F., "Gedanken-experiments on Sequential Machines.", Automata Studies, Annals of Mathematical Studies, 34, pp.129-153, 1956