

Causality 프로토콜 명세서

김태인@DSPLab

tikim@dsp.inha.ac.kr

1. 요약

인과를 기반으로 구축한 시나리오 설계 프로토콜인 Causality의 명세사항을 기술한다.

2. 문서 정보

- 최종 수정 일자: 2020년 5월 3일
- 프로토콜 버전: 0.0.2

목차

| | |
|------------------------------|----|
| Causality 프로토콜 명세서 | 1 |
| 1. 요약 | 1 |
| 2. 문서 정보..... | 1 |
| 3. 설계 이론..... | 4 |
| 4. 객체 설명..... | 5 |
| 4.1. 배역 (Role) | 5 |
| 4.2. 행동 (Action)..... | 5 |
| 4.3. 퍼셉트론 (Perceptron) | 6 |
| 4.4. 원인 (Cause) | 7 |
| 4.5. 선택지 (Option) | 8 |
| 4.6. 캡션 (Caption)..... | 8 |
| 4.7. 장면 (Scene) | 9 |
| 4.8. 시나리오 (Scenario)..... | 9 |
| 4.9. 맥락 (Context) | 10 |
| 5. XML 구현 방법..... | 11 |
| 5.1. 배역 (Role) | 11 |
| 5.2. 행동 (Action)..... | 11 |
| 5.3. 퍼셉트론 (Perceptron) | 12 |
| 5.4. 원인 (Cause) | 12 |
| 5.5. 선택지 (Option) | 13 |
| 5.6. 캡션 (Caption)..... | 13 |
| 5.7. 장면 (Scene) | 14 |

| | | |
|------|----------------------|----|
| 5.8. | 시나리오 (Scenario)..... | 14 |
| 5.9. | 맥락 (Context) | 14 |

3. 설계 이론

본 프로토콜의 설계는 사회적 교류를 시뮬레이션 하는 프로그램들은 어떻게 설계하는 것이 효율적인가에 대한 고찰에서부터 시작한다.

예를 들면, 맥락의 흐름은 고정되어 있어 사용자가 선택지를 고르는 것이 영향을 미치지 않는 환경을 만드는 것은 비교적 단순하다. 하지만 사용자의 선택이 동적으로 맥락의 흐름에 계속해서 영향을 미치는, 비교적 현실적인 구조를 흉내내기 위해선 고려해야 할 것이 많아진다. 가능한 한 객체지향적으로 끌어내어보면 인과론에 기반해 원인과 결과 이 두 가지 관점으로 모든 선택과 행동을 감싸주면 진전이 있지 않을까 생각이 들었다.

선택과 행동은 Action이라는 범주로 생각해보자. 그리고 전체 시나리오는 자잘한 장면의 조합으로 생각해보면 두 개념은 각각 Scenario와 Scene 클래스로 생각할 수 있을 것이다. 그리고 시나리오라는 흐름을 떠다니는, 마치 배와 같은 존재를 현재의 상태를 담고 있는 맥락(Context)으로 간주할 수 있다. 하나의 장면은 사용자에게 제공할 Action들을 담고 있다. 여기에서 제공한 Action을 사용자가 수행하면, 이들에 플래그를 남겨 다른 장면에서도 이전에 사용자가 Action을 수행했는지 여부를 참조하도록 만들어 두면 인과관계의 뼈대를 만들 수 있을 것이다.

장면은 다음에 이어질 장면들을 가리키고 있어야 한다. 그리고 그 중 어디로 가야 하는지 맥락에게 알려주는 결정 과정도 담고 있으면 유기적인 스토리를 구현할 수 있을 것이다. 예를 들어, 시나리오 A에서 B가 아닌 C로 넘어가기 위해 지금까지 사용자가 액션 1,3,4를 행해야 했다면 단순한 조건검사로 구현 가능하다. 하지만 만약 각 액션별로 중요도를 부여하고 싶다면? 1,3번을 안 했지만 4번을 했거나 4번을 안 했지만 1,3번을 했으면 진행되게 하고싶다면? 이런 경우엔 불린(AND, OR, NOT) 형식의 조건 검사보다는 퍼셉트론(Perceptron) 형태로 검사를 구현하는 것이 더 좋을 수 있다. 1,3번의 가중치를 0.3으로 두고 4번의 가중치를 0.6으로 두면 위에서 말한 조건을 퍼셉트론 연산으로 쉽게 구현할 수 있다. 이를 통해 오픈월드처럼 단방향 스토리 진행이 아닌 것도 구현할 수 있을 것이다.

선행 연구 조사의 일환으로 비주얼노벨 개발엔진인 RenPy 를 살펴보았는데, 시나리오나 선택지, 행동이 서브루틴, 조건문 같이 프로그래머 친화적으로 서술되도록 되어있어서 콘텐츠 제작자와 프로그래머 간의 역할 분리가 힘들 것으로 예상되었다. 이에 본 프로토콜은 XML과 퍼셉트론 기반 조건연산에 기반을 두어 최대한 전문적인 지식이 없어도 시나리오를 구성할 수 있는 것을 목표로 하였다.

4. 객체 설명

본 프로토콜을 구성하는 객체들에 관해 설명한다.

4.1. 배역 (Role)

배역 객체는 시나리오에 출연하는 등장인물들을 표현한다. 이 때 배역의 위치(Position)을 정하는 것으로 주인공, 조연, 나레이터 등을 구분할 수 있다.

| 속성 이름 | 타입 | 필수여부 | 설명 |
|--------------|------------------|------|--|
| SerialNumber | String | O | 배역의 일련번호. |
| Name | String | O | 배역의 이름 |
| Position | Enum Position | O | 배역의 위치(주연-Leader, 조연-Supporter, 단역-Extra, 나레이터-Narrator) |
| Description | String | O | 배역에 대한 설명 |

표 1. Role 클래스 명세

4.2. 행동 (Action)

행동 객체는 고유한 행동을 표현해야 한다. 이를 위해 행동 일련번호와 행동 설명, 행동 주체와 같은 속성을 가져야 하며, 해당 행동이 수행되었는지 여부도 기록할 수 있어야 한다. 경우에 따라 행동에 따른 값이 Value로 대표될 수 있다.

| 속성 이름 | 타입 | 필수여부 | 설명 |
|--------------|---------|------|-------------------|
| SerialNumber | String | O | 행동의 일련번호. |
| Description | String | O | 행동에 대한 설명 |
| Value | String | X | 행동의 내용 |
| ActorName | String | O | 행동 주체의 이름 |
| IsPerformed | Boolean | O | 행동이 수행되었는지에 대한 여부 |

표 2. Action 클래스 명세

4.3. 퍼셉트론 (Perceptron)

퍼셉트론은 조건을 나타낼 수 있는 객체이다. x_n 은 조건으로 이용할 행동이나 캡션, 혹은 장면이며, 행동일 경우엔 IsPerformed, 캡션과 장면은 Displayed 여부에 따라 0 혹은 1이 된다. w_n 은 조건에 곱해질 가중치이며 b 는 전역적으로 더해질 편향이다. 이러한 연산은 아래 그림과 수식으로 표현할 수 있다.

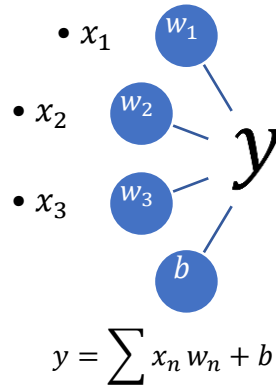


그림 1. 퍼셉트론의 구조 및 수식

| 속성 이름 | 타입 | 필수여부 | 설명 |
|--------------|--------------|------|----------------------|
| SerialNumber | String | O | 퍼셉트론의 일련번호. |
| Description | String | O | 퍼셉트론에 대한 설명 |
| Inputs | List<String> | O | 입력할 객체의 일련번호를 담은 리스트 |
| Weights | List<Double> | O | 가중치를 담은 리스트 |
| Bias | Double | O | 편향 값 |

4.4. 원인 (Cause)

원인 객체는 캡션이나 장면을 표시할 지에 대한 여부를 결정해준다. 원인은 다수의 퍼셉트론을 가지고 있으며, 퍼셉트론의 연산 결과에 따른 다음 캡션 또는 장면에 대한 정보도 포함한다.

| 속성 이름 | 타입 | 필수여부 | 설명 |
|--------------|--------------|------|---------------------------------|
| SerialNumber | String | O | 원인의 일련번호. |
| Description | String | O | 원인에 대한 설명 |
| Perceptron | String | O | 퍼셉트론의 일련번호 |
| Next | List<String> | O | 퍼셉트론의 결과에 따라 이동할 캡션 또는 장면에 일련번호 |
| Thresholds | List<Double> | O | Next를 결정할 문턱 값 |

원인 조건의 판단에 대한 설명을 다음 페이지에서 계속한다.

- 만일 Next 의 원소가 A, B 이고, Thresholds 가 0.5 하나라면, 퍼셉트론의 연산 결과가 0.5 미만이면 A가 다음에 이동할 객체가 되고 0.5 이상이면 B가 다음에 이동할 객체가 된다.
- 만일 Next 의 원소가 A, B, C 이고, Thresholds 가 0.3, 0.6 이라면, 퍼셉트론의 연산 결과가 0.3 미만이면 A, 0.3 이상 0.6 미만이면 B, 0.6 이상이면 C가 다음에 이동할 객체가 된다.

위와 같이 Thresholds 의 개수는 항상 Next 의 개수보다 하나 적어야 한다.

4.5. 선택지 (Option)

선택지 객체는 복수의 행동을 포함하며 다중선택 여부와 선택된 행동의 일련번호도 갖고 있다.

| 속성 이름 | 타입 | 필수여부 | 설명 |
|---------------|--------------|------|----------------------|
| SerialNumber | String | O | 선택지의 일련번호. |
| Description | String | O | 선택지에 대한 설명 |
| Actions | List<String> | O | 행동의 일련번호를 담은 리스트 |
| IsMultiple | Boolean | O | 다중선택이 가능한지 여부 |
| ChosenActions | List<String> | X | 선택된 행동의 일련번호를 담은 리스트 |

표 3. Option 클래스 명세

4.6. 캡션 (Caption)

캡션 객체는 장면의 내용을 대화 혹은 독백 형태로 서술하며, 서술의 주체와 다음에 따를 캡션의 일련번호를 가지고 있다. 이 때 캡션이 나오기 위한 원인을 만족하지 못하면 캡션을 표시하지 않고 원인이 가리키는 캡션으로 넘어간다.

| 속성 이름 | 타입 | 필수여부 | 설명 |
|--------------|---------|------|------------------------|
| SerialNumber | String | O | 캡션의 일련번호. |
| Dialogue | String | O | 캡션 내용 |
| HasCause | Boolean | O | 캡션이 나오기 위한 조건이 있는지의 여부 |
| Cause | String | X | 원인의 일련번호 |
| SpeakerName | String | O | 캡션 주체의 이름 |
| Option | String | X | 캡션과 함께 보여줄 선택지의 일련번호 |
| NextCaption | String | O | 다음 캡션의 일련번호 |
| Displayed | Boolean | O | 캡션이 표시되었는지의 여부 |

표 4. Caption 클래스 명세

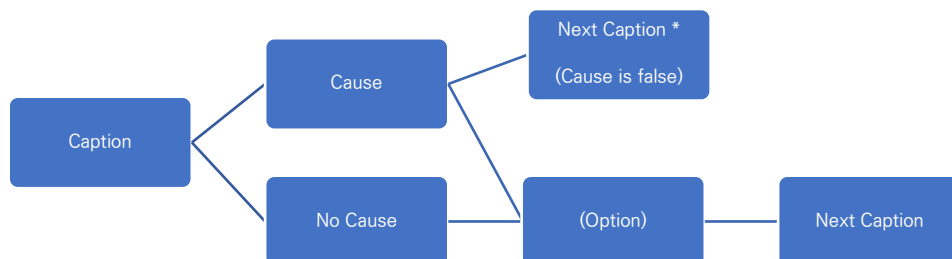


그림 2. 캡션의 조건 및 선택지 유무에 따른 흐름

4.7. 장면 (Scene)

장면 객체는 하나의 캡션을 가지고 있으며, 캡션이 포함하는 선택지를 통해 사용자에게 행동을 선택 받도록 한다.

| 속성 이름 | 타입 | 필수여부 | 설명 |
|--------------|---------|------|------------------------|
| SerialNumber | String | O | 장면의 일련번호. |
| Description | String | O | 장면에 대한 설명 |
| Caption | String | O | 캡션의 일련번호 |
| HasCause | Boolean | O | 장면이 나오기 위한 조건이 있는지의 여부 |
| Cause | String | X | 조건의 일련번호 |
| Displayed | Boolean | O | 장면이 표시되었는지의 여부 |

표 5. Scene 클래스 명세

4.8. 시나리오 (Scenario)

시나리오 클래스는 하나 이상의 장면 객체를 가지고 있으며, 여러 개의 장면을 갖고 있다면 보유한 순서대로 표시를 시도하게 된다.

| 속성 이름 | 타입 | 필수여부 | 설명 |
|--------------|--------------|------|------------------|
| SerialNumber | String | O | 시나리오의 일련번호. |
| Description | String | O | 시나리오에 대한 설명 |
| Scenes | List<String> | O | 장면의 일련번호를 담은 리스트 |
| Displayed | Boolean | O | 시나리오가 표시되었는지의 여부 |

표 6. Scenario 클래스 명세

4.9. 맥락 (Context)

맥락 클래스는 현재 위치한 시나리오, 장면, 캡션을 가지고 있다.

| 속성 이름 | 타입 | 필수여부 | 설명 |
|-----------------|--------|------|-----------------|
| SerialNumber | String | O | 맥락의 고유번호 (GUID) |
| CurrentScenario | String | X | 현재 시나리오 번호 |
| CurrentScene | String | X | 현재 장면 번호 |
| CurrentCaption | String | X | 현재 캡션 번호 |

표 7. Context 클래스 명세

5. XML 구현 방법

본 단원에서는 앞에서 소개한 각 객체를 XML로 기술하는 방법을 기술한다. 각 예시는 올바른 작성 예를 나타낼 뿐이고, 본 문서에 작성된 예시만으로는 작동이 가능한 완전한 예를 만들 수 없다.

5.1. 배역 (Role)

```
<Role>
  <SerialNumber>R000</SerialNumber>
  <Name>나레이터</Name>
  <Position>Narrator</Position>
  <Description>해설 캡션을 담당</Description>
</Role>
<Role>
  <SerialNumber>R001</SerialNumber>
  <Name>김민수</Name>
  <Position>Leader</Position>
  <Description>시나리오 주인공</Description>
</Role>
<Role>
  <SerialNumber>R002</SerialNumber>
  <Name>점원</Name>
  <Position>Supporter</Position>
  <Description>카페 점원</Description>
</Role>
```

이 때 Position 타입은 Enum으로 구현되었고 Leader, Supporter, Extra, Narrator 가 정의되어 있다고 전제한다.

5.2. 행동 (Action)

```
<Action>
  <SerialNumber>A008</SerialNumber>
  <Description>커피-크기 Venti</Description>
  <Value>Venti</Value>
  <ActorName>김민수</ActorName>
  <IsPerformed>false</IsPerformed>
</Action>
<Action>
  <SerialNumber>A009</SerialNumber>
  <Description>커피-크기 직원에게 문의</Description>
  <ActorName>김민수</ActorName>
  <IsPerformed>false</IsPerformed>
</Action>
```

```

<Action>
  <SerialNumber>A010</SerialNumber>
  <Description>커피를 매장에서 마신다</Description>
  <Value>매장에서 마실게요.</Value>
  <ActorName>김민수</ActorName>
  <IsPerformed>false</IsPerformed>
</Action>

```

5.3. 퍼셉트론 (Perceptron)

```

<Perceptron>
  <SerialNumber>P000</SerialNumber>
  <Description>커피 사이즈 선택</Description>
  <Inputs>
    <Input>A006</Input>
    <Input>A007</Input>
    <Input>A008</Input>
    <Input>A009</Input>
  </Inputs>
  <Weights>
    <Weight>0.5</Weight>
    <Weight>0.5</Weight>
    <Weight>0.5</Weight>
    <Weight>0.0</Weight>
  </Weights>
  <Bias>0</Bias>
</Perceptron>

```

5.4. 원인 (Cause)

```

<Cause>
  <SerialNumber>U000</SerialNumber>
  <Description>사이즈 선택여부</Description>
  <Perceptron>P000</Perceptron>
  <Nexts>
    <Next>C008</Next> <!-- 사이즈 물어보기 -->
    <Next>C007</Next> <!-- 사이즈 말함 -->
  </Nexts>
  <Thresholds>
    <Threshold>0.5</Threshold>
  </Thresholds>
</Cause>

```

5.5. 선택지 (Option)

```
<Option>
  <SerialNumber>0000</SerialNumber>
  <Description>커피 메뉴 선택</Description>
  <Actions>
    <Action>A000</Action> <!-- 커피-아메리카노 선택 -->
    <Action>A001</Action> <!-- 커피-카페라떼 선택 -->
    <Action>A002</Action> <!-- 커피-녹차 선택 -->
    <Action>A003</Action> <!-- 커피-초코라떼 선택 -->
  </Actions>
  <IsMultiple>false</IsMultiple>
</Option>
```

5.6. 캡션 (Caption)

```
<Caption>
  <SerialNumber>C001</SerialNumber>
  <Dialogue>점심시간이라 사람이 많은 스타벅스 안, 한참 기다렸다. 여전히 민수 뒤에는 사람들이 많다.</Dialogue>
  <HasCause>false</HasCause>
  <SpeakerName>나레이터</SpeakerName>
  <NextCaption>C002</NextCaption>
  <Displayed>false</Displayed>
</Caption>
<Caption>
  <SerialNumber>C002</SerialNumber>
  <Dialogue>점원: 어서오세요. 주문하시겠어요?</Dialogue>
  <HasCause>false</HasCause>
  <SpeakerName>점원</SpeakerName>
  <NextCaption>C003</NextCaption>
  <Option>0000</Option>
  <Displayed>false</Displayed>
</Caption>
<Caption>
  <SerialNumber>C003</SerialNumber>
  <Dialogue>{0} 주세요.</Dialogue>
  <HasCause>false</HasCause>
  <SpeakerName>김민수</SpeakerName>
  <NextCaption>C004</NextCaption>
  <Displayed>false</Displayed>
</Caption>
```

5.7. 장면 (Scene)

```
<Scene>
  <SerialNumber>S000</SerialNumber>
  <Description>카페(요청) - 주문</Description>
  <Caption>C000</Caption>
  <HasCause>false</HasCause>
  <Displayed>false</Displayed>
</Scene>
```

5.8. 시나리오 (Scenario)

```
<Scenario>
  <SerialNumber>E000</SerialNumber>
  <Description>카페(요청) 시나리오</Description>
  <Scenes>
    <Scene>S000</Scene>
  </Scenes>
  <Displayed>false</Displayed>
</Scenario>
```

5.9. 맥락 (Context)

```
<Context>
  <SerialNumber>D87C6088-F5AC-426A-AF34-99BE0656CD9A</SerialNumber>
  <CurrentScenario>E000</CurrentScenario>
  <CurrentScene>S000</CurrentScene>
  <CurrentCaption>C001</CurrentCaption>
</Context>
```