

# Structured Causal Graph에 기반한 이벤트 전개 방법의 개발

지세진<sup>1)</sup> 우영욱 황원택 최운논 박종희  
경북대학교 전자공학과

{husty, lenna, xordl, undon}@palgong.knu.ac.kr, jhpark@ee.knu.ac.kr

## Modeling of Event Development based on the Structured Causal Graph

Se-Jin Ji<sup>1)</sup> Young-Wook Woo Won-Taek Hwang Un-Don Choi Jong-Hee Park  
Dept. of Electronic Engineering, Kyungpook National University

### 요약

지금까지 가상현실을 이용한 여러 가지 시스템들이 제안되어 왔다. 하지만 이러한 시스템들은 사용자에게 몰입감을 줄 수 있는 예측하기 힘든 다양한 상황을 제공하기보다는 미리 정해진 시나리오를 따라 고정된 형태의 흐름을 가지는 문제점이 있었다. 이를 해결하기 위해 가상세계 내에서의 오브젝트의 행동이나 이벤트의 전개를 위한 여러 가지 방법들이 제안되어왔다. 하지만 이 방법들 역시 방대한 탐색 공간이나 한정된 범위내에서만 자율적인 움직임이 가능한 점 등의 문제점을 가지고 있다. 본 논문에서는 이를 해결하기 위하여 Causality에 기반한 이벤트의 전개모형을 제안한다. 이를 위해 본 논문은 먼저 frame구조를 이용하여 정형화한 Structured Causal Graph를 제안하고, 구성되어진 Structured Causal Graph를 이용하여 이벤트를 전개해나가는 방법을 제시한다.

### 1. 서론

지금까지 제작되어진 대다수의 컴퓨터를 기반으로 한 교육용 시스템들은 해당 교육분야의 전문가들의 지식을 바탕으로 구축되어진 데이터베이스에 있는 지식들을 피교육자에게 전달하는 방식으로 개발되어 왔다. 하지만 이러한 일방적인 전달 형태의 교육용 시스템에 의해 피교육자가 얻을 수 있는 지식은 매우 추상적이고 한정된 형태의 것에서 벗어나지 못한다[1]. 이러한 문제점을 극복하기 위해 최근에는 기존의 텍스트 기반의 환경에서 벗어나, 멀티미디어 데이터와 가상현실에 기반을 둔 형태의 교육용 시스템들이 개발되고 있다. 하지만 가상현실에 기반한 대다수의 교육용 시스템들은 규칙이나 룰에 기반하여 이벤트가 전개되기보다는 저작시 구성되어진 시나리오의 정해진 흐름을 따라 진행되도록 구성되어 있어 피교육자에게 몰입감을 주기 위한 다양한 상황을 제공하지 못하고, 피교육자와 시스템 사이에 제한된 형태의 상호작용만을 제공한다[3].

이러한 문제점을 해결하기 위해서는 미리 정해진 시나리오에 의한 이벤트의 발생이 아니라 규칙과 룰에 따른 이벤트의 전개 방법이 필요하다. 이런 이벤트의 전개를 위해 지금까지 Rule-Based 모델, 뉴럴 네트워크 모델, Predefined 모델 등의 여러 가지 방법들이 제안되었으나 탐색공간의 방대함 등 여러 가지 문제점으로 인해 제한된 범위의 응용에서만 사용되었을 뿐 범용적으로 사용되지는 못하였다[2]. 본 논문에서는 기존의 이벤트 전개 방법들의 문제점을 해결하기 위해 causality에 기반한 이벤트의 전개 방법을 제안하고자 한다.

causality에 기반한 이벤트의 전개는 가상세계에서 일어나는 변화에 의해 새로운 이벤트가 발생하는 방식을 취함으로써 정해진 시나리오를 따라서 전개되는 기존의 Path 모델의 단점을 보완한다. 이를 위해 Structured Causal graph를 제안하고, 등장하는 오브젝트와 환경에 따라 구성되어진 Causal graph에서 적합한 노드만을 택하는 방법을 제안함으로써 기존의 이벤트 전개 모델들이 가지던 방대한 탐색공간으로 인한 문제점들을 적절히 제어한다.

본 논문은 먼저 지금까지의 이벤트 전개 모델들의 문제점을 살펴보고, causality에 기반한 이벤트의 전개를 위해 Causal graph를 정형화한다. 다음으로 구성되어진 Causal graph를 이용하여 이벤트를 전개하는 방식에 대해 알아보고 마지막으로 이러한 연구에 대한 결론을 내리고 앞으로 향후 연구방향에 대해 기술한다.

### 2. 관련연구

#### 2.1 객체의 행위 모델

컴퓨터 애니메이션 분야에선 객체가 고정된 환경 하에서 정해진 경로를 따라 행동하는 Path model이 지배적으로 사용되어 왔다. 하지만 이러한 모델은 지속적인 변화가 일어나고 있는 동적인 환경 하에서 발생하는 다양한 상황과 예측할 수 없는 시점에서 발생하는 이벤트들에 대해 적응적으로 대응하지 못한다는 결점이 있다. 이러한 결점을 극복하기 위해 Sensor-Effector Model, Rule-Based Model, Predefine Model, Relation-Based Model등이 연구되어 지고 있다.[2] Sensor-Effector Model은 Sensor와 Effector를 각각 감지요소와 구동요

소로 하고 이들 사이의 제어 메커니즘으로 neural network를 채택한 모델이다. 이 모델이 효용을 가지기 위해서는 제어 메커니즘인 neural network의 구조에 대한 깊은 이해가 필요하나 아직까지 이에 대한 명확한 검증이 이루어지지 않고 있다. Rule-based model은 제어 메커니즘으로 행위규칙들로 구성되어진 rule base를 이용한다. 하지만 가능한 모든 동적 행위들을 커버하기 위해서는 방대한 규모의 Decision Tree를 구성하고 탐색해야 한다는 문제점을 가지고 있다. Predefined 모델의 경우는 어떤 행동을 시작하기 이전에 미리 결정된 상황에만 적용이 가능하다는 문제점을 가진다. Relation-Based Model은 두 개의 관련된 object사이의 relation을 제어 메커니즘으로 사용하여 Rule-base model의 방대한 탐색 공간 문제를 해결하려 하였다. 하지만 이 Relation-based model은 object의 행위가 저작자가 임의로 선택한 제한된 범위 안에서 제한되고 object의 행위가 자율적으로 발생하지 못한다는 문제가 있다.

### 3. Structured Causal Graph의 정형화

#### 3.1 Structured Causal Graph

causality에 기반한 이벤트의 전개 모델은 기본적으로 cause/effect를 나타내는 노드와 그들 사이의 causal relation을 나타내는 edge로 이루어진 cause/effect쌍들로 구성되어진 Causal graph를 이용하여 전개되어진다.

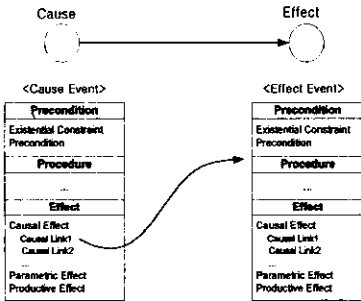


그림 1 Cause/Effect Pair

Structured Causal graph를 형성하고 있는 노드들은 각각 하나의 이벤트를 나타내며 이 이벤트들은 frame형태로 구성되어진다. 또 이 노드들간의 causal relation은 이벤트 frame의 Causal Effect 슬롯을 통해 나타내어진다.

하나의 이벤트를 나타내는 frame은 이벤트가 일어나기 위한 전제 조건을 나타내는 Preconditions, 가상환경에서의 구체적인 행위를 기술한 Procedure, 그리고 이 이벤트의 결과로 나타나는 Effect의 세 부분으로 나뉜다. Preconditions는 다시 이 이벤트가 발생하기 위해 반드시 존재해야 하는 object들을 나타내는 existential constraint 슬롯과 각 object의 attribute들이 가져야 하는 속성 값의 범위 등을 나타내는 productive effect 역시 일종의 Causal Effect로 볼 수 있으나 이는 새로운 오브젝트의 생성이라는 특수한 경우이므로 따로 나누어 고려하기로 한다. 본 논문에서는 이 3가지의 Effect 중 Causal effect를 이용하여 하나의 이벤트가 새로운 이벤트를 발생시키는 방법에 중점을 두고자 한다.

### 3.2 Causal Effect

한 이벤트의 frame에서 Causal Effect 슬롯은 이 이벤트에 의해 새롭게 발생하게 되는 이벤트들을 나타낸다. Causal Effect는 여러개의 Causal link들로 구성되어진다.

하나의 Causal link는 이 이벤트에 의해 새롭게 발생하는 effect event의 주체가 되는 object를 나타내는 target object, 이 Causal link가 가리키는 이벤트가 발생하기 위한 최소한의 조건을 표시하는 Essential Condition, 그리고 그 effect로 발생하게 된 event를 가리키는 Effect Event 슬롯을 가지게 된다.

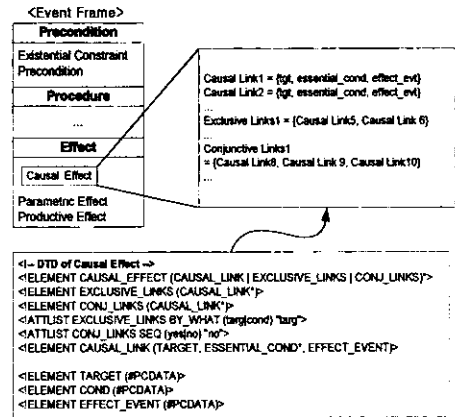


그림 2 Causal Effect Slot의 구조 및 DTD 정의

기본적으로 이러한 Causal link들은 disjunctive한 관계를 가진다. 하지만 몇몇 Causal link들은 conjunctive한 관계를 가지기도 하며 exclusive한 관계를 가지는 경우도 있다.

conjunctive한 Causal link들은 반드시 함께 발생해야만 하는 이벤트들을 가리키기 위한 것이다. conjunctive한 관계를 가지는 Causal link들은 sequential하게 실행되어야 하는 것과 단순한 conjunctive관계를 가지는 경우로 나뉘어진다.

exclusive한 관계를 가지는 causal link들은 그 중 오직 하나만이 선택되어야 하는 경우이다. Causal link의 결과로 나타나는 어떤 event들은 그 주체가 되는 object들에 의해 exclusive한 관계를 가질 수 있다. 또한 어떤 조건이나 속성값의 범위에 따라 서로 이런 exclusive한 관계가 형성되어질 수 있다.

disjunctive, conjunctive, exclusive한 관계의 Causal link들은 각각 그 성질에 따라 object의 명령 리스트[4]에 저장되어 지는 방식이 달라지게 된다. 이에 관한 내용은 4절에서 다루기로 한다.

### 4. Structured Causal Graph를 이용한 Event 전개

Structured Causal graph를 이용한 event의 전개에서 새로운 event가 발생하게되는 구동요소는 가상환경에서 어떤 지역적인 변화이다. 즉, 가상환경내에 있는 오브젝트에 대한 이벤트가 발생하여 가상환경내에 변화를 가져오게 되면 이 변화에 의해 어떤 오브젝트의 새로운 이벤트가 발생하게 되는 것이다. 이 경우 새롭게 발생하게 되는 이벤트의 주체가 되는 오브젝트는 원래 그 변화를 초래한 오브젝트 자신일 수도 있고 다른 오브젝트가 될 수도 있다.

하나의 이벤트 의한 가상환경 내의 변화가 새로운 이벤트의 발생을 가져오게 하기 위해서 전술했던 이벤트의 Frame 구조 중 Causal

Effect 슬롯을 이용한다. Causal Effect 슬롯은 해당 이벤트가 새롭게 발생시킬 수 있는 가능성이 있는 이벤트들을 가리키고 있는 복수의 Causal Link들로 구성되어있다. 진정한 사이버스페이스를 구축하기 위해서는 Causal graph가 매우 치밀하고 복잡하게 형성되어야 하므로 어떤 이벤트의 Causal Effect 슬롯은 매우 많은 수의 Causal link를 가지게 된다. 결과적으로 Causal graph mechanism에 기반한 이벤트의 전개를 위해서는 매우 방대한 탐색 공간을 다루어야만 하게 된다. 이러한 문제의 해결을 위해 본 논문에서는 Causal link 슬롯에 새롭게 일생하는 이벤트의 주체가 되는 오브젝트를 나타내는 Target Object 슬롯과 이 Causal Link에 의해 이벤트가 발생하기 위한 최소한의 조건을 나타내는 Essential Condition 슬롯을 이용하여 발생할 수 있는 가능성이 있는 Causal Graph의 범위를 줄인다. 즉, 현재의 장면에 나타나 있지 않은 오브젝트에 대한 Causal Link와 현재 가상세계의 상태가 만족시키지 못하는 Essential Condition을 가지는 Causal Link를 탐색공간에서 제외시키는 것이다. 이를 통해 방대한 Causal Graph중 일부분을 선택하여 Rule-based 시스템의 방대한 탐색공간 문제를 해결한다. 하지만 이러한 Causal link의 pruning은 해당 노드 뿐 아니라 그 노드를 통해 transitive하게 연결되어지는 노드들마저 탐색공간에서 제거해 버리는 문제점이 있다. 하지만 이러한 transitive한 연결은 형성되어진 Causal graph에서는 매우 예외적인 경우이며 잘 구축되어진 Causal graph라면 이 노드들의 경로가, 제거되어진 노드를 통하는 하나만이 아니라 복수개의 경로를 가지게 되므로 이는 그다지 큰 문제가 되지 않는다.

위의 두 가지 제약 조건을 이용하여 축소되어진 Causal graph를 따라 새롭게 발생하게 될 이벤트들이 선택되어 지게 된다. 이 때 선택되어진 이벤트들은 선택되어진 시점에서 즉시 발생하게 되는 것이 아니라 그 주체가 되는 오브젝트의 명령 리스트에 추가되어 진다. 명령 리스트에 추가되어진 이벤트들은 실행되는 시점에서 event의 frame 구조에서 Precondition 슬롯에 명시된 조건을 검사하게 되고 이를 통해 Causal graph를 이용하여 찾아낸 결과가 실행 시점의 조건에 따라 실행되어질 수도 있고 실행되어지지 않을 수도 있게 되므로 다양한 상황의 표현이 가능해 진다.

Structured Causal graph를 이용하여 새롭게 발생되어질 이벤트를 선택하여 해당 오브젝트의 명령 리스트에 추가하는 과정에서 disjunctive, conjunctive, exclusive한 관계를 가지는 Causal link들에 대해 각각 다른 처리 과정이 필요하다. disjunctive한 관계의 Causal link들의 경우는 단순히 해당 오브젝트의 명령 리스트에 추가하기만 하면 되지만 conjunctive한 Causal link들의 경우 이 conjunctive한 Causal link들이 sequential한지의 여부에 따라 처리 방법이 달라진다. sequential하게 일어나야 하는 두 개의 Causal link들의 경우 이들을 순서에 맞게 명령 리스트에 삽입하는 것이 필요하다. exclusive한 관계의 Causal link들은 그 중 하나만이 선택되어지는 작업이 필요한데 명령 리스트에 삽입하기 전에 미리 exclusive한 Causal link들의 두가지 선택 조건에 따라 하나의 Causal link가 선택되어져야 한다.

**5. 적용 예**

본 논문에서 제안한 Causality에 기반한 이벤트의 전개에 있어 이벤트를 발생시키는 요소가 되는 것은 어떤 이벤트를 발생시키는 원인이 되는 또다른 이벤트의 발생이다. 이를 보이기 위해 이벤트의 frame들을 노드로 하는 Causal graph를 형성하고, Causal Link들이 가지는 Target과 Essential Condition을 이용하여 이 Causal graph를 적절하게 pruning하고(그림3) 이렇게 선택되어진 Node들 중 새롭게 발생할 이벤트를 가지키는 노드들을 각 오브젝트의 명령리스트에 추가하고 그 이벤트 노드에 정의되어진 precondition을 만족하는 이벤트들을 실행시키는 시스템을 구현한다.

하나의 적용 예로 '비'라는 이벤트의 발생이 다른 오브젝트들의 자율적인 이벤트들을 발생시키는 경우를 살펴보자. 현재의 장면에 나타

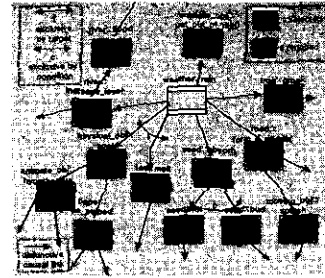


그림 3. 이벤트 프레임의 노드로 하여 형성되어진 Causal Graph

내어진 오브젝트들이 agent, seed, tree등이 있다면 구성되어진 causal graph중 이들 오브젝트에 종속되어진 이벤트 frame들을 제외한 나머지 노드들은 고려대상에서 제외시킨다. 이렇게 남은 노드들 중에서 '비'라는 이벤트의 causal effect slot이 가리키고 있는 노드들은 해당 오브젝트의 명령 리스트에 추가되어지며 명령 리스트에서 실행되어지는 시점에서 이벤트의 precondition slot에 명시된 조건을 검사하여 실행여부를 판단하게 된다.

구현은 Microsoft Visual C++ 6.0과 MFC library를 이용하여 각각의 이벤트 frame은 정의되어진 dtd를 따르는 xml document로 표현되어진 다.

**6. 결론 및 향후 연구 방향**

본 논문에서는 가상환경에서의 자율적이고 논리에 맞는 이벤트의 전개를 위해 Causality에 기반을 둔 이벤트 전개 방식을 제시하였다. frame 구조를 이용하여 이벤트를 정형화된 형태로 나타내고, 이 정형화된 이벤트 frame들을 노드로 하는 Causal graph를 형성하여, 가상세계에서의 어떤 변화가 다른 이벤트를 자동적으로 발생시킬 수 있는 방법을 제안하였다. 또 현재 가상세계에 존재하는 오브젝트나 수치적인 조건등에 의해 선택되어진 Causal graph의 일부분을 고려대상으로 하여 새롭게 발생될 이벤트를 찾아가게 되므로 방대한 탐색공간을 찾아야만 하던 기존의 시스템의 문제를 일부 해결하였다.

본 논문에서는 오브젝트에 종속적인 이벤트에 대한 연구만이 진행되었다. 하지만 모든 이벤트가 오브젝트에 종속적인 것은 아니다. 한 예로 오브젝트들의 충돌과 같은 것을 들 수 있다. 앞으로 이러한 복수의 오브젝트 사이에서 발생하는 이벤트들을 전개하기 위한 방법에 대한 연구가 진행되어야 할 것이며, 또한 오브젝트의 명령 리스트를 관리함에 있어 인터럽트에 대한 처리나 병렬적으로 수행되어야 하는 이벤트들에 대한 처리 등에 관한 연구가 필요할 것이다.

**참고문헌**

[1] Yukihiro Matsubara, Seiji Toihara, Yuichiro Tsukinari, Mitsuo Nagamachi, "Virtual Learning Environment for Discovery Learning and Its Application on Operator Training", IEICE Transaction on Information & Systems, Vol. E80-D, No.2, February 1997  
 [2] Hanqiu Sun, "A Relation-Based Model for Animating Adaptive Behavior in Dynamic Environments", IEEE Transaction on Systems, Man, and Cybernetics - Part A : Systems and Humans, Vol. 27, No. 2, March 1997  
 [3] Park, J., "A logical simulation of discretionary events in spatio-temporal context", Tech. report #3, AIMM lab., Kyungpook Nat' Univ., 2000  
 [4] 박형근, "동적 환경에서의 자율 에이전트 행위 설계 및 구현", 경북대학교 석사학위논문, 1999