

# 협력적 가상환경을 위한 내용기반 동적 이벤트 통지시스템

장진윤<sup>0</sup>    탁진현    이세훈\*    왕창중  
인하대학교 전자계산공학과, \*인하공업전문대학 전자계산기과  
{ellio, tak, }@selab.cse.inha.ac.kr, seihoon@true.inhatc.ac.kr,  
c.jwangse@inha.ac.kr

## Content-Based Dynamic Event Notification System for Collaborative Virtual Environment

Jin-Yoon Chang<sup>0</sup>    Jin-Hyun Tak    Sei-Hoon Lee\*    Chang-Jong Wang  
Dept. of Computer Science & Engineering, Inha University  
\*Dept. of Computer Engineering, Inha Technical College

### 요 약

이 논문에서는 협력적 가상환경에서 이벤트의 내용에 기반한 동적 이벤트 통지 시스템을 제안한다. 제안된 시스템은 이벤트 송신자와 수신자 사이에 투명성을 제공하며, 다른 시스템에 영향을 주지 않고, 이벤트 수신자를 동적으로 추가, 삭제할 수 있는 노티파이어 모델을 구성한다. 시스템은 기본적인 이벤트 형태들을 정의하고, 다른 환경을 구성하고 있는 특정 응용에 독립적으로 동작하도록 하기 위해서 이벤트들을 XML로 표현한다. 본 시스템은 XML을 사용하여 이벤트를 표현하기 때문에 발생한 이벤트를 로그데이터에 저장할 수 있고, 로그 데이터를 사용하여 이벤트들을 검색하고 필터링할 수 있도록 하여 시스템의 효율성을 향상시켰다. 또한 사용자로 하여금 관심있는 이벤트들을 등록할 수 있게 하여 필터링의 효율을 높였으며 XML 기반 애플리케이션뿐만 아니라 다른 가상환경의 애플리케이션들 사이의 상호 작용을 가능하게 하는 이벤트 통지 시스템을 설계하였다.

### 1. 서론

최근에는 컴퓨팅 능력의 발달과 네트워크의 향상으로 인하여 인터넷 기술이 급속하게 발전함에 따라 VRML을 비롯한 가상현실에 대한 관심이 증가하고 있다[1]. 특히 협력적 가상 환경에서의 공동 작업을 위한 참여자간의 상호작용에 대한 연구가 활발히 진행되고 있으며, 이러한 상호 작용을 지원할 수 있는 이벤트 전달방식이 필요하게 되었다[2, 3]. 현재 대부분의 가상현실 시스템에서는 독자적인 시스템과 이벤트 전달방식을 사용하고 있으며, 애플리케이션에 종속적인 이벤트 형식을 정의하고 있다. 따라서, 가상환경의 확장과 다른 환경과의 통합을 고려하고 있지 못하는 문제점이 발생한다. 그러므로, 협력적 가상환경에서 발생하는 기존의 이벤트들을 수용할 수 있고, 애플리케이션들간에 발생하는 이벤트들을 효율적으로 필터링하는 기법이 필요하다. 특히 가상환경에서의 공동작업이 상호 영향을 받는 경우나 발생하는 이벤트를 공유하여야 하는 경우에서 서로 전송되는 이벤트들의 양을 줄이고 이러한 이벤트들을 효율적으로 검색하기 위해 로그데이터화할 필요가 있다. 따라서 본 논문은 협력적 가상환경에서 XML을 기반으로 한 이벤트 통지 시스템을 제안한다. 제안된 시스템은 이벤트 송신자와 수신자간의 정보

를 분리시키고 수신자의 동적인 이벤트 등록을 제공하는 이벤트 통지(Event notification) 모델을 구현하는 노티파이어(notifier)를 설계한다.

이러한 노티파이어는 이벤트의 송신자와 수신자가 빈번하게 변경되는 가상환경에 적합하다. 또한, 제안된 시스템은 내용 기반의 구조화와 자기-기술적인 정보 전달의 표준인 XML을 사용하여 이벤트의 형식을 정의함으로써, 특정 애플리케이션 도메인에 독립적으로 작용한다. 표준화된 이벤트는 가상 현실 애플리케이션뿐만 아니라 XML 기반의 모든 애플리케이션간의 상호작용에 이용할 수 있고 복잡한 변환 없이 가상 환경에 적용할 수 있다.

### 2. 관련연구

가장 간단한 이벤트 전달 방식은 이벤트 송신자가 발생하는 이벤트들을 이벤트 송신자들에게 전달해 주는 방식이다. 하지만 이벤트 수신자가 모든 이벤트 송신자들을 참조하고 있어야 하는 단점이 있다. 이러한 단점을 극복하기 위한 이벤트 전달 방식은 다음과 같다[4,5].

#### • 중재자 모델(Mediator model)

이 방식은 이벤트 수신자와 송신자 사이에 중재자가 있어서 이벤트 송신자들이 이벤트를 중재자에게만 이벤트를 송신하면 중재자가 그 이벤트들을 이벤트 수신자들에게 전달해 주는 것이다. 이 방식의 단점은 이벤트 수신자가 늘어나는 경우 중재자가 관리하는 이벤트 수신자들도 추가

해줘야 하는데 있다[5,6].

• 관찰자 모델(Observer model)

이 방식은 이벤트 수신자들이 관심있는 이벤트들을 등록할 수 있게 하고, 이벤트 수신자들은 이벤트가 발생하면 등록한 이벤트들만을 수신하게 된다. 이 관찰자 방식은 이벤트 수신자들이 증가하더라도 다른 참여자들에게 영향을 주지 않고 시스템을 확장할 수 있는 장점이 있지만, 이벤트 수신자들이 송신자들의 정보를 모두 알고 있어야 하는 단점이 있다[5,6].

• 이벤트 통지 모델(Event Notifier model)

이벤트 통지 방식은 위 두가지 방식의 장점만을 취한 것으로 증재자 방식처럼 중앙에 이벤트를 증재해주는 증재자를 두어 이벤트를 처리해주고, 관찰자 방식과 같이 등록 시스템을 두어서 이벤트 수신자들을 추가하고 삭제할 수 있게 하였다. 그림 1은 이벤트 통지 모델에 대한 UML 표기법을 보여준다[5].

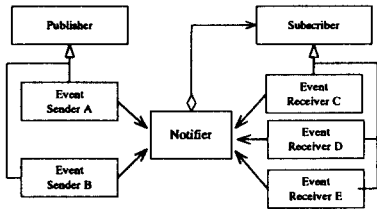


그림 1. UML 표기법에 따른 이벤트 통지 모델

3. 이벤트 통지 시스템 설계

이 장에서는 협력적 가상환경에서의 이벤트 통지 시스템을 설계한다. 설계되는 이벤트 통지 시스템은 이벤트 송신자(publisher)와 이벤트 수신자(subscriber)사이 에 투명성을 제공하여 복잡한 이벤트 전달 경로를 단순화시키고, 공유 객체나 특정 정보에 대한 이벤트 수신을 동적으로 추가, 제거 및 필터링 기능을 수행한다. 또한, 내용기반의 이벤트를 정의하여 기존의 도메인 종속적인 애플리케이션들에 대한 상호 작용성을 제공한다. 설계한 이벤트 통보 시스템은 그림2와 같다.

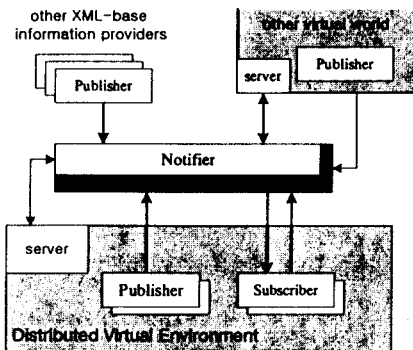


그림 2. 이벤트 통지 시스템 구조

3.1 event specification

가상환경에서의 이벤트를 수용하기 위한 기본적인 이벤트 타입을 다음과 같이 정의한다.

• Action Event

가상환경에서 객체 상태의 변화나 능동적인 행위가 일어났을 때 발생하는 이벤트들이다. 이 이벤트들은 가상환경의 일관성유지와 관련된 것으로, 동기화 정도에 따라 아래와 같이 정의된다.

- Independent interaction event : 일반 객체의 이동이나 아바타의 위치변화와 같이 참여자 독립적으로 발생하는 이벤트들로 다른 참여자들에게 즉시 영향을 미치지 않아 최근의 이벤트가 이전의 이벤트를 대체할 수 있다. 하지만 일관성을 위해서 다른 클라이언트들에게 반영되어야 한다.
- Shared interaction event : 이벤트를 발생시키는 데 있어서, 다른 클라이언트들에게 즉시 영향을 미치는 이벤트들로 Independent event 들보다 우선적으로 반영되어야 한다. 예를 들면, 공동 작업을 하는데 있어서 발생하는 이벤트들은 바로 참여자들에게 반영되어야 한다.

• Information event

가상환경의 동기화에 관련되지 않는 이벤트이다. 이것은 일관성 유지를 위해 즉시 통보되지 않아도 된다. 그러나, 반드시 전달되어야 하는 이벤트이다. 가상환경 내부 또는 외부에 보내지는 mail 형태의 데이터를 보낼 때, 또는 시스템으로부터 이전의 로그데이터 등을 요구할 때 발생된다.

위 이벤트들의 자료구조는 다음 그림 3 과 같다.

```

Event {
  Type = INDPENENT | SHARED | INFORMATION
  {
    Session ;
    ObjectID;
    AvatarID;
    NotifyRange = Interaction | System | Session |
                userDef [Outer | AvatarIDs;
    TimeStamp;
    Content { .... }
  }
}
    
```

그림 3. 이벤트 자료구조

그림 4는 통지를 위한 Shared Interaction Event 의 자료구조를 XML 문법으로 기술하는 예를 보인다. 이것은 "Ellio "라는 이름의 아바타가 공동 작업의 대상인 객체, "CoWork"를 <vec3fx> 의 어트리뷰트의 값에 따라 이동시킨 이벤트이다. <Content> 의 서브엘리먼트는 이 이벤트에 의해 변화가 일어나야 하는 VRML 의 노드이름과 값을 나타내기 위해 사용된다.

XML로 기술한 이벤트는 가상공간의 참여자가 발생시킨 이벤트에 대한 로그 정보를 데이터베이스화하기 쉽고 내용기반의 필터링을 가능하게 한다. 또한, 이러한 정보는 가상공간에서 참여자의 행위 패턴이나 작업정보를 기록하고, 발생한 이벤트를 검색하는 데에 유용하다.

```

<?xml version="1.0" encoding="UTF-8"?>
<Event type="SHARED">
  <Session>Field1</Session>
  <AvatarID>Ellio</AvatarID>
  <ObjectID>CoWork</ObjectID>
  <NotifyRange>InteractionRange
</NotifyRange>
  <TimeStamp>05:12:04</TimeStamp>
  <Content>
    <node type="translation">
      <vec3fx value="10","-2","5"/>
    </node>
  </Content>
</Event>
</xml>
    
```

그림 4. XML 을 사용한 이벤트 형식의 예

3.2 노티파이어(Notifier)

설계된 노티파이어는 이벤트의 등록과 이벤트 발생 다른 가상 환경과 같은 외부 서비스에서 발생한 이벤트들을 XML 로 변환해주는 변환기(Converter)와 이러한 이벤트들을 필터링하여 그 양을 줄이는 필터(Filter)로 구성된다.

퍼블리셔로부터 이벤트가 노티파이어에 전달되면 subscribers에게 전달되기 전에 변환기와 필터를 거쳐 전파된다. 그림 5는 이벤트가 통지 되는 과정에서의 XML 변환기의 동작과 필터링의 등록, 그리고 그 적용시점을 보여준다.

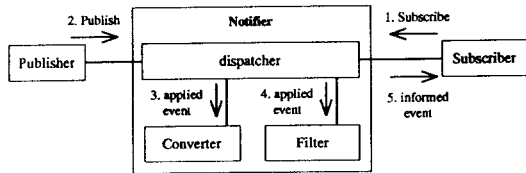


그림 5. 이벤트 통지 시스템의 Collaboration Diagram

변환기는 다른 가상 환경의 시스템에서 발생한 이벤트일 경우, 서로의 상호작용을 위하여 이벤트를 XML로 변환해주는 모듈이다.

필터는 발생한 이벤트들을 subscriber 들이 등록한 이벤트들을 검색하여 필터링한 후, 해당되는 이벤트들만을 전송한다. 이 때 노티파이어는 미리 이벤트가 발생한 가상환경의 서버에 접속하여 이벤트의 전송 범위를 알도록 한다.

등록된 필터는 애플리케이션을 관리하는 서버(또는 컨트롤러)에 의해 적용되는 Default Filter 와 각각의 애플리케이션에서 제공되는 사용자 인터페이스를 통해 등록할 수 있는 Personalized Filter 로 구분된다. 기본필터는 Action Event 에 대하여 가상공간에서 발생하는 프리젠테이션 이벤트에 대한 필터링을 수행한다.

information Event 에 대하여는 가상공간에서의 CSCW인 경우 작업의 룰이나 정책에 의해 적용될 수 있다. 이벤트 필터링은 단순히 사용자의 요구에 따르는 필터링을 수행할 뿐만아니라, 가상환경에서 프리젠테이션을 위해 발생하는 수 많은 이벤트를 필

터링함으로써, 전체 가상공간에서 전파되는 이벤트의 양을 줄일 수 있다는 것이 중요하다.

또한, Personalized filter 는 사용자가 관심 있는 이벤트 수신에 반영하기 위하여 사용된다. 보통의 경우, Action event 에 대한 등록은 시스템에 의해 자동적으로 필터링되어지고, information event 의 경우에 참여자의 관심 정도에 따라 개인 필터가 적용된다.

4. 결론

이 논문에서는 협력적 가상환경에서 이벤트의 내용에 의한 동적 이벤트 통지 시스템을 제안하였다. 제안된 시스템은 이벤트 송신자와 이벤트 수신자사이에 투명성을 제공하여 복잡한 이벤트 전달 경로를 단순화시켰다. 또한, 가상 환경을 관리하는 서버측의 공간 관리 방식을 통지 시스템에 등록함으로써 관리 방식에 독립적인 이벤트 전파를 가능하게 하였고, 노티파이어에 해당 애플리케이션에 대한 인터페이스를 등록시켜 XML 형태의 이벤트로 변환이 이루어질 수 있다. 인터페이스는 기존의 가상 환경과의 통합이나 그 환경과의 상호 연동을 제공하고, 가상환경 밖에서 전달되는 XML 문서를 변환 없이 참여자의 가상공간으로 전달할 수 있다.

또한, 공유객체나 특정 정보에 대한 이벤트 수신을 시스템에 영향을 주지 않고 동적으로 추가되거나 제거할 수 있으며, 동적인 필터링 기능을 제공하였다. 서브스크라이버는 특정 이벤트에 대한 필터를 노티파이어에 등록함으로써, 노티파이어를 통하여 전달되는 이벤트는 personalized 할 수 있다. 또한, XML 로 표현된 이벤트는 <tag>의 검색 및 변환을 이용하여 이벤트 내용에 대한 필터링을 가능하게 한다.

5. 참고 문헌

- [1]. "The Virtual Reality Modeling Language", International Standard ISO/IEC 14772-1:1997
- [2]. Olof Hagsand, "Interactive multiuser Ves in the DIVE system", IEEE Multimedia Magazine, 3(1):30-39, 1996.
- [3]. Funkhouser, ThomasA. "RING: A Client -Server System for Multi -User Virtual Environments", Computer Graphics(1995 SIGGRAPH Symposium on Interactive 3D Graphics), pp 85-92, 1995.
- [4]. Soren Brandt, Anders Kristensen, "Web Push as an Internet Notification Service," online White paper, 1997, <http://keryxsoft.hpl.hp.com/doc/ins.html>
- [5]. S. Gupta, J. Har tkopf, S. Ramaswamy, "Event Notifier: A Pattern for Event Notification", in Java Report Magazine, July 1998.
- [6]. Gamma et al., "Design Patterns : Elements of Reusable Object-Oriented Software", Addison-Wesley, 1997.