

---

# CEP 기반 온라인 게임 악용 패턴 모니터링 방법

## Abuse Pattern Monitoring Method based on CEP in On-line Game

---

노창현  
중부대학교 게임학과

Chang Hyun Roh(chroh@joongbu.ac.kr)

---

### 요약

본 연구에서는 CEP(Complex Event Processing) 기법을 사용하여 온라인 게임에서 사용자 악용을 실시간 감지하기 위한 효율적인 방법을 제시한다. CEP는 대규모 정보시스템에서 발생하는 복잡한 이벤트 패턴을 발견하는 기법이다. 본 연구에서는 게임 사용자가 게임 서버에 접속하여 게임을 수행하는 과정에서 발생하는 이벤트들을 관찰하면서 규칙에 위배되는 행위를 검출하기 위하여 CEP 기법을 사용한다. 사용자들이 자주 악용할 가능성이 있는 이벤트 패턴을 미리 복합 이벤트로 설정하여 CEP 엔진에 등록해 두고, CEP 엔진은 게임 서버에서 발생하는 이벤트를 실시간에 필터링하여 사용자 악용을 감시하는 방법이다.

■ 중심어 : | 온라인 게임 감시 | 사용자 악용 감시 | CEP |

### Abstract

Based on a complex event processing technique, an abuse pattern monitoring method is developed to provide a real-time detection. CEP is a technique to find complex event pattern in a massive information system. In this study, the events occurred by game-play are observed to be against the rules using CEP. User abuse patterns are pre-registered in CEP engine. And CEP engine monitors user abuse after aggregating the game data transferred by game logging server.

■ keyword : | Online Game Monitoring | Abuse Pattern Monitoring | CEP |

---

## 1. 서론

1990년대 후반 국내 인터넷이 급속히 보급되면서 인터넷 관련 업체들이 대거 등장하고 있다. 이에 따라 다양한 형태의 온라인 서비스가 진행되고 있다. 특히 커뮤니티 포털, 온라인 게임, 온라인 쇼핑 등의 성장이 매우 두드러진다.

온라인 콘텐츠 서비스의 특징은 24시간에 걸쳐 이루어진다는 것이고 사용초기부터 종료까지 사람의 손을 거치지 않고 사용자의 요구에 대응해야 한다는 것이다.

그러므로 24시간 사용자에게 좋은 품질의 서비스가 지속되고 있는지에 대해 감시하고 분석하는 것은 매우 중요한 일이다.

온라인 서비스 장애는 서비스 시스템 결함에 의해 생기는 문제도 있지만 서비스 이용자들의 고의적인 또는 우연적인 행위에 의해 문제가 발생하는 경우가 빈번하다. 이러한 점을 방지하고 관리하기 위해서는 수많은 사용자들의 방대한 서비스 기록을 분석하거나 모니터링을 해야 하지만, 이를 위해서는 엄청난 시간과 인력이 소모되기 마련이다. 이 때문에, 온라인 서비스를 효

울적으로 관리하고 감시해 주는 도구에 대한 필요성은 꾸준히 제기되고 있다. 그러나 일반적으로 1) 조사하여야 할 데이터양이 방대하고, 2) 사용자의 사용 행태가 예측이 불가능할 정도로 매우 다양하며, 3) 온라인 서비스의 지속적인 업그레이드와 함께 감시 도구의 특징도 매번 함께 업그레이드되어야 하는 등의 어려움 때문에, 온라인 서비스 감시 도구는 서비스 업체에 따라 자체 개발에 의존하고 있으며 상용화 제품은 많지 않은 실정이다.

따라서 온라인 서비스의 감시 도구는 넘쳐나는 데이터 속에서도 이러한 상황을 가급적 신속(Early Warning)하게 알려주는 기능을 갖추어야 한다. 또한 비정상적인 상황이 나타나더라도 이것이 타당한 원인이 있기 때문에 그런 것인지, 아니면 불법적인 방법에 의해 나타난 현상인지를 판단할 수 있는 방법도 온라인 서비스 감시 장치는 갖추고 있어야 한다.

그러므로 본 연구에서는 CEP(Complex Event Processing)엔진을 사용하여 온라인 서비스 중에 온라인 게임 사용자들의 악용(Abuse) 패턴을 감시하는 방법을 제안한다.

온라인 게임이란 여러 사용자들이 인터넷으로 게임 서버에 접속하여 주어진 목표를 달성하기 위하여 대결하거나 자신의 캐릭터를 성장시키는 게임이다. 대표적인 온라인 게임 유형인 MMORPG(Massively Multiplayer Online Role Playing Game)의 경우, 여러 사용자들이 인터넷 서버의 가상 공간에서 공동체를 이루고 괴물 등을 사냥하고 게임머니(game money)를 벌어들이고 아이템을 획득하거나 구매하면서 자신의 캐릭터를 성장시킨다[1].

이러한 게임 개발업체들은 개발된 게임 소프트웨어를 여러 대의 온라인 서버에 설치하고 주기적으로 업데이트하면서 게임 서비스를 제공한다[2]. 온라인 게임은 개발 프로그램이 방대하고 진행 과정이 매우 복잡하기 때문에, 게임 출시 이후에도 여러 가지 버그가 발생하는 경우가 많아 지속적으로 발견하여 수정해야 한다. 그러나 게임 개발자들이 인식하기 전에 발견된 버그를 사용자들이 자신에게 유리하도록 이용하는 경우가 종종 있으며, 이를 어뷰징(abusing)이라고 부른다. 어뷰징

방법이 사용자에게 의해서 발견되면, 지인들을 통하여 게임 이용자들 사이에 급속도로 확산되는 양상을 보인다. 만약 어뷰징에 대한 대응이나 프로그램 수정이 늦어지는 경우에는 정상적인 사용자들이 상대적인 피해를 입게 되어 게임에 대한 신뢰도가 하락하게 될 뿐만 아니라, 심각한 경우에는 사용자 이탈로 인한 게임의 존폐에 영향을 미치기도 한다. 이 때문에, 얼마나 빨리 프로그램 버그를 발견하고 사용자 악용을 감시하고 방지하느냐가 게임 서비스의 신뢰도 유지에 매우 중요한 이슈이다.

그런데, 온라인 게임에서는 사용자 수가 워낙 많고 게임 진행 과정도 매우 복잡하기 때문에 실시간 감시 기능을 구현하기가 쉽지 않다. 또한 게임 실행 결과가 주기적으로 저장되는 데이터베이스도 분석하기 매우 방대하며, 매일 대규모 데이터가 누적되기 때문에 설령 분석한다고 하더라도 즉각적인 조치를 취하기가 어렵다.

본 연구에서는 CEP (Complex Event Processing, 복잡 이벤트 처리) 기법을 사용하여 온라인 게임에서 사용자 악용을 실시간 감지하기 위한 효율적인 방법을 제시한다. CEP는 대규모 정보시스템에서 발생하는 복잡한 이벤트 패턴을 발견하는 기법이다. 본 연구에서는 게임 사용자가 게임 서버에 접속하여 게임을 수행하는 과정에서 발생하는 이벤트들을 관찰하면서 규칙에 위배되는 행위를 검출하기 위하여 CEP 기법을 사용한다. 사용자들이 자주 악용할 가능성이 있는 이벤트 패턴을 미리 복합 이벤트로 설정하여 CEP 엔진에 등록해 두고, CEP 엔진은 게임 서버에서 발생하는 이벤트를 실시간에 필터링하여 사용자 악용을 감시하는 방법이다.

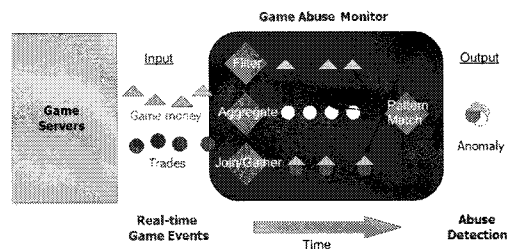


그림 1. 온라인 게임 실시간 감시 틀

그림1에서 보는 바와 같이 본 연구에서 제시하는 방법은 CEP를 이용하여 거의 실시간으로 게임 사용자의 악용 상황을 발견할 수 있기 때문에, 온라인 게임의 신뢰성을 확보하고 안정적인 서비스를 제공하는 데 도움이 될 수 있다.

## II. 관련 기술

### 1. 이벤트 처리 기법[3]

대규모 시스템에서 이벤트 처리 기법은 크게 단순 이벤트 처리(Simple Event Processing), 이벤트 스트림 분석(Event Stream Processing), 복합 이벤트 처리(Complex Event Processing)로 구분할 수 있다.

단순 이벤트 처리는 특정한 데이터가 주어진 조건을 만족하면 정해진 규칙을 단순히 처리하는 방법이며, 이벤트 스트림 처리(ESP)는 계속적으로 발생하는 이벤트 스트림에서 특정한 패턴이 발견되면 정해진 규칙을 처리하는 기법이다. 주식, 환율, 유가 등의 경제 데이터나 RFID, 센서 데이터, 수요나 판매 데이터 등이 이벤트 스트림 처리의 대상이 될 수 있다.

복합 이벤트 처리(CEP)는 대규모 정보시스템에서 여러 가지 이벤트 스트림이 복합적으로 발생할 때, 복합 이벤트 패턴을 정의하여 원하는 상황을 추출하기 위한 기법이다. 복합 이벤트 처리는 일반적으로 이벤트 스트림 기법을 포괄하는 의미이며, 이벤트 연관관계, 이벤트 인과관계, 이벤트 계층화, 이벤트 추상화 등의 개념을 활용한다.

최근에 기업정보시스템 등 다양한 시스템이 결합된 환경에서 복합적인 상황을 감지하고 위험 관리 및 의사 결정을 지원하기 위하여 CEP 기법이 많이 도입되고 있다. 특히, 서비스 지향적 아키텍처(SOA: Service Oriented Architecture)의 약점을 보완하기 위하여 제시되고 있는 이벤트 주도적 아키텍처(EDA: Event Driven Architecture)에서 이벤트 중심의 시스템 통합 환경의 실시간 상황 인지 및 조기 경보를 위하여 CEP가 도입되고 있다.

CEP의 적용 예를 살펴보면, 주가 변동 감지, 신용카드

부정 사용, 비밀번호 도용, 여러 가지 센서 감지, 서비스 대응 속도 등 다양한 분야의 이벤트 감지 및 처리를 위하여 시도되고 있다.

본 연구에서는 MMORPG와 같이 대규모 사용자가 동시 접속하여 진행되는 복잡한 온라인 게임 환경에서 사용자들의 어뷰즈 패턴을 실시간에 검출하기 위하여 CEP를 도입하였다.

### 2. CEP 기술

CEP 기술은 데이터베이스 기술과 자주 비교되고 있다. 데이터베이스는 대규모 시스템에서 자주 업데이트 되는 데이터 및 정보를 체계적으로 저장하여 필요한 데이터를 추출할 수 있게 지원하는 기술이다. 반면에 CEP 기술은 대규모 시스템에서 지속적으로 발생하는 이벤트 정보를 모니터링 하면서 필요한 이벤트 패턴을 추출할 수 있게 지원하는 기술이다.

데이터베이스 관리를 위하여 지원 시스템으로 DBMS를 사용한다면, CEP 처리를 위해서는 CEP 엔진을 사용한다. CEP 엔진은 여러 시스템에서 발생하는 이벤트를 지정하고, 추출하고자 하는 이벤트 패턴을 등록하면, 여러 가지 이벤트 스트림들을 정제(filtering), 합산(aggregation), 수집(gather), 결합(join)한 후 패턴 매칭을 통하여 원하는 이벤트 패턴의 발생을 감지하는 기능을 제공한다.

데이터베이스 관리를 위한 표준언어로 SQL (Structural Query Language)가 사용된다면, CEP 처리를 위해서는 EPL (Event Processing Language)를 사용한다. EPL은 실제로 SQL과 유사한 문법 구조를 사용하지만, 아직 표준화된 단계는 아니며 각 CEP 엔진에 따라 약간 차이가 있다. EPL은 원하는 이벤트 스트림들 중에서 원하는 조건에 충족되는 상황을 추출하는 구문을 정의하는데, SQL처럼 SELECT-FROM-WHERE 절을 기본적으로 사용한다. 본 연구에서 활용하는 Esper[4] 엔진에서 사용하는 EPL의 기술 방법을 살펴보면 아래와 같다.

- 시간 구간 (Data Window)

```
select * from TradeEvent.win:time(30 seconds)
```

30초 동안의 TradeEvent를 관찰한다.

- 이벤트 정제 (Filtering)

```
select * from TradeEvent(symbol='IBM').win:time(30 seconds)
```

30초 동안 'IBM'의 TradeEvent를 관찰한다.

- 이벤트 합산 (Aggregation)

```
select sum(price) from TradeEvent(symbol='IBM').win:time(30 seconds)
```

30초 동안 'IBM'의 TradeEvent에서 가격의 합산을 계산한다.

- 이벤트 결합 (Join)

```
select * from TradeEvent(symbol='IBM').win:time(30 seconds) as te, SettleEvent(symbol='IBM').win:time(30 seconds) as se where te.orderId = se.orderId
```

30초 동안 'IBM'의 TradeEvent와 SettleEvent를 결합하여 보여준다.

- 삽입 (Insert-Into) 및 출력 (Output-Every)

```
insert into SMA /* Simple Moving Average */ select symbol, avg(price) as avgPrice from Tick.win:time(5 days) group by symbol output all every 10 minutes
```

최근 5일 동안의 Tick에서 종목(symbol)과 평균 주가(avgPrice)를 계산하여 SMA 이벤트로 10분 단위로 저장한다.

- 패턴 매칭 (Pattern Matching)

```
every e1 = SMA(symbol='GOOG') -> ( timer:interval(60 min) and not SMA(symbol='GOOG', avgPrice < e1.avgPrice) )
```

SMA의 GOOG 이벤트에서 60분 간격의 평균 주가보다 (10분 단위의)평균 주가가 더 높은 경우를 모두 추출한다.

### III. 온라인 게임 감시 과정

#### 1. 어뷰즈 패턴 (Abuse Pattern)

온라인 게임의 어뷰즈 패턴은 게임의 종류에 따라 다를 수 있지만 MMORPG의 경우 게임머니, 아이템 취득과 관련하여 어뷰즈 사례가 많다. 이와 같은 어뷰즈 형태를 나름대로 정형화 할 수 있다면, 게임 관리자는 이처럼 잠재적인 어뷰징을 탐지할 수 있는 패턴을 지정함으로써 사용자들의 어뷰징을 실시간에 감지하여 사용자들의 어뷰징 위험에 빠르게 대응할 수 있다.

게임 어뷰징의 예를 살펴보면, 하나의 아이템을 두 사용자가 거의 동일한 시간과 장소에서 동시에 취득함으로써 두 명이 각각 아이템을 획득하는 경우가 있다. 이는 게임 공간 내에서의 아이템 총개수를 필요 이상으로 늘림으로써 아이템의 가치를 하락시키는 결과를 가져올 수 있다. 이러한 어뷰징을 방지할 수 있는 패턴은 특정 공간의 아이템 획득 시 총 개수를 확인하여 비정상적으로 개수가 증가하는지를 확인하는 것이다.

또 다른 예를 들면, 사용자 간의 아이템 거래시에 한 사용자가 아이템 구입 후에 비정상 종료 등을 악용하여 지불을 하지 않는 경우이다. 이러한 경우는 아이템 획득의 전후 시점을 확인하여 게임 머니의 증감수준과 아이템 가격을 비교하는 방법을 사용할 수 있다.

#### 2. 감시 과정

본 연구에서 제시하는 어뷰즈 감시 과정은 [그림 2]와 같이 수행된다. 먼저 관리자는 어뷰즈 패턴을 미리

정의하여 CEP 엔진에 등록하고, 실시간에 사용자들의 게임 실행 이벤트를 수집하여 CEP 엔진에게 전달하면, 어뷰징을 감지하여 관리자에게 보고하는 과정으로 구성된다.

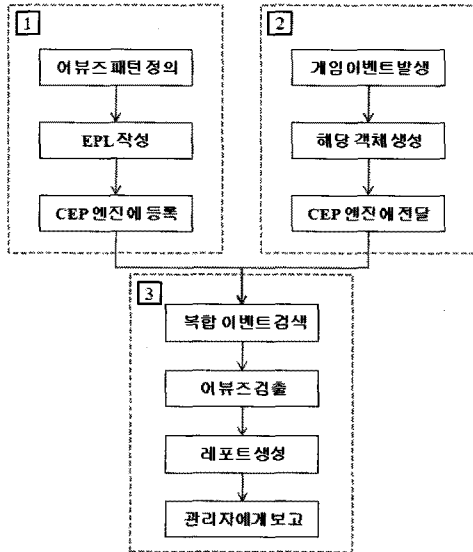


그림 2. 온라인 게임 감시 과정

### 2.1 어뷰징 패턴 정의 및 등록

해당 게임에서 자주 발생하고 게임 서비스에 영향을 많이 미치는 어뷰징 패턴을 미리 정의하여 CEP 엔진에 등록시킨다.

- 거래 후 게임 머니의 불일치나, 아이템의 비정상적인 증가 등 검출하고자 하는 어뷰징을 추출할 수 있는 패턴을 정의한다.
- 사용자들의 게임 머니, 아이템, 경험치, 등급 등이 감시 대상이 된다.
- 어뷰징이 자주 발생하는 시점이나 장소, 거래 등을 지정한다.
- 불일치 또는 무결성이 깨어지는 조건을 정의한다.
- 정의된 어뷰징 패턴을 CEP 엔진이 해석할 수 있는 이벤트 기술 언어인 EPL (Event Processing Language)로 작성한다.
- 작성된 EPL을 CEP 엔진에 등록시킴으로써 복합적인 이벤트 스트림을 감시할 준비를 한다.

### 2.2 실시간 게임 이벤트 수집

게임 서버에서 발생하는 이벤트들을 어뷰징 모니터에 전달하기 위하여, 관련되는 이벤트 객체를 생성하고 CEP 엔진에 전달한다.

- 게임 이벤트들은 여러 게임 서버에서 직접 전달받는 것이 아니라, 로깅 서버로부터 간접적으로 전달 받는다.
- CEP 엔진에서 이벤트를 처리하기 위하여 해당 이벤트를 표현할 수 있는 이벤트 객체를 생성한다.
- 생성된 이벤트 객체는 복합 이벤트를 추출하는 CEP 엔진에게 전달된다.

### 2.3 실시간 어뷰징 발견 및 보고

미리 정의된 어뷰징 패턴과, 로깅 서버로부터 전달된 이벤트 스트림을 비교하여 어뷰징이 발생하는지를 게임 관리자에게 보고한다.

- CEP 엔진은 지속적으로 발생하는 다양한 이벤트 스트림이 존재할 때, 미리 정의된 EPL에 해당되는 복합 이벤트가 발생하는지 감시한다.
- 게임 서버에서 전달되는 이벤트 객체들을 CEP 엔진이 처리하여, 게임 관리자가 미리 정의한 어뷰징 패턴에 해당되는 것들을 보고서에 기록한다.
- 게임 관리자는 실시간에, 또는 주기적으로 보고서를 확인하여 사용자 어뷰징을 확인하고 대응 조치를 취한다.

## IV. 온라인 게임 시스템 구조

일반적으로 온라인 게임들은 여러 대의 게임 서버를 운용하는데, 이 게임 서버들은 개별적으로 사용자들의 게임을 실행시키고 상황을 전제한 후, 중요한 사용자 데이터 및 현황 정보를 로깅 서버를 통하여 데이터베이스에 저장한다[2]. 게임 서버에서 발생하는 이벤트들은 실시간의 상세한 정보를 포함하고 있지만, 게임 서버에 저장되는 정보는 불필요할 정도로 세부적인 데이터들이 많을 뿐만 아니라, 게임 서버의 이벤트를 직접 전달 받는 것은 게임 서버의 속도 및 성능에 영향을 줄 수 있다.

그러므로 본 연구에서는 게임 서버로부터 직접 이벤트를 수집하는 것이 아니라, 게임 서버들이 주기적으로 로깅 서버에게 전달하는 이벤트를 분석함으로써, 게임 서버의 실행과는 완전히 분리된 구조로 이벤트를 수집한다.

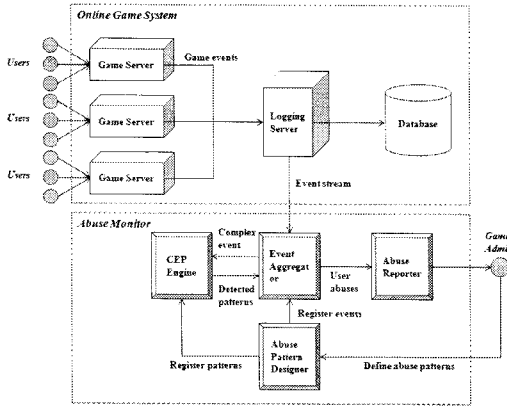


그림 3. 온라인 게임 감시 시스템 구조

[그림 3]은 본 연구의 온라인 게임 감시 시스템의 구조를 보여준다. 온라인 게임 서버는 주기적으로 중요한 게임 사용자 데이터를 저장하기 위하여 로깅 서버로 데이터를 전달하는 데, 감시 시스템은 이 로깅 서버로부터 이벤트를 수집하여 이벤트 처리를 작동한다.

게임 관리자는 미리 Abuse Pattern Designer를 통하여 감시 대상이 되는 어뷰즈 패턴을 정의하면, 이는 실시간 이벤트를 수집하는 Event Aggregator에 이벤트를 등록하고, CEP 엔진에 추출할 이벤트 패턴을 등록한다. CEP Engine은 Event Aggregator로부터 전달받은 이벤트들을 감시하여 어뷰즈를 추출할 후, Abuse Report를 생성하여 관리자에게 전달한다.

## V. 프로토타입 설계 및 구현

제시된 온라인 게임 감시 방법을 테스트하기 위하여 아래와 같은 프로토타입 시스템을 설계하여 구현 중에 있다. CEP 엔진으로는 Esper[4]를 사용하였으며, 개발 언어는 Java를 사용하였다. 개발된 프로토타입 시스템

의 주요 컴포넌트를 설명하면 아래와 같다.

- 게임 로그 시뮬레이터 (Game Log Simulator)
  - 사용자 n명과 아이템 m개를 가정하고 거래를 가상으로 수행한다.
  - 일부 거래에 확률적으로 abuse를 포함시킨다.
  - 거래 시 거래이벤트(TradingEvent)와 골드이벤트(GoldEvent)를 발생시킨다.
- 게임 이벤트 (TradingEvent, GoldEvent)
  - 모니터링할 거래이벤트과 골드이벤트를 정의한다.
  - 거래이벤트는 거래수행내역을 인스턴스화한다. (buyerID, sellerID, itemName)
  - 골드이벤트는 사용자의 골드양을 인스턴스화한다. (userID, gold)
- 이벤트 리스너 (Game Listener)
  - 거래이벤트와 골드이벤트를 모니터링한다.
  - 이벤트 패턴을 지정하여 해당되는 패턴을 추출한다.
- 테스트 (TestGameEvent)
  - 시뮬레이션을 실행시키고, 리스너를 생성하여 패턴을 추출하는 테스트를 실행한다.
- 이벤트 로그 해석기 (Event Log Parser)
  - 실제 게임에 직접 테스트해볼 수 없기 때문에, 실제 게임업체에서 획득한 게임로그 파일을 파싱하여 실제상황과 유사하게 이벤트를 발생시키도록 로그를 해석한다.
- 이벤트 어뷰즈 패턴 (Game Abuse Patterns)
  - 감시할 사용자 어뷰즈를 감지할 유형들을 다양하게 설정하여, 시스템에 반영시킨다.
- 게임 리포터 (Game Reporter)
  - 모니터링 결과를 보여주기 위한 감시보고서를 작성하여 보여준다.

```

package org.game.monitor;

public class TradingEvent {
    private String buyerID;
    private String sellerID;
    private String itemName;

    public TradingEvent(String buyerID,
        String sellerID, String itemName) {
        this.buyerID = buyerID;
        this.sellerID = sellerID;
        this.itemName = itemName;
    }

    public String getBuyerID() {
        return buyerID;
    }

    public String getSellerID() {
        return sellerID;
    }

    public String getItemName() {
        return itemName;
    }
}
    
```

그림 4. 게임 이벤트 설계 예시

```

package org.game.monitor;

import com.espertech.esper.client.UpdateListener;
import com.espertech.esper.event.EventBean;

public class GameListener implements
UpdateListener {
    public void update(EventBean[] newEvents,
        EventBean[] oldEvents){

        EventBean event = newEvents[0];
        System.out.println("avg=" + event.get("avg(gold)")
            + " userID=" + event.get("userID"));
    }
}
    
```

그림 5. 게임 리스너의 이벤트 감시 예시

```

// get engine instance
EPServiceProvider epService =
    EPServiceProviderManager.getDefaultProvider();
String expression = "select avg(gold),userID
    from org.game.monitor.GoldEvent.win:time(1 sec)
    group by userID"
EPStatement statement =
    epService.getEPAAdministrator().createEPL(expression);

// set up statement
GameListener listener = new GameListener();
statement.addListener(listener);

// generate events
GameSimulator gs = new GameSimulator(1000);
    
```

그림 6. CEP 엔진 생성 및 EPL 등록 방법

## VI. 결론 및 향후 연구

현재까지는 온라인서비스 감시의 어려운 점을 효과적으로 해결해 줄 수 있는 다양한 감시 기법에 대한 선행 기술들을 비교 평가하였다. 본 연구는 CEP(Complex Event Processing, 복잡 이벤트 처리)기법을 해결 방안으로 채택하였다. CEP는 대규모 정보시스템에서 발생하는 복잡한 이벤트 패턴을 발견하는 기법으로서, 본 연구에서는 게임 사용자가 게임 서버에 접속하여 게임을 수행하는 과정에서 발생하는 이벤트들을 관찰하면서 규칙에 위배되는 행위를 검출하기 위하여 CEP 기법을 사용하였다. CEP의 기본 개념 및 관련 라이브러리 등에 대한 활용에 대해서는 간단한 예제를 직접 개발하여 검증함으로써 기술 습득을 마쳤다. 게임사로부터 제공된 실제 온라인서비스 데이터를 일부 분석하여, 그 속에 내재된 이벤트 패턴을 분석하고 있으며, 여기에서 도출된 패턴은 복합 이벤트로 설정하여 CEP 엔진에 등록해 두고, CEP 엔진은 게임 서버에서 발생하는 이벤트를 실시간에 필터링하여 온라인서비스의 불법적 사용을 방지할 수 있다.

향후 계획은 프로토타입 온라인서비스 감시 도구의 구현을 마치고, 현재 서비스 중인 또는 온라인서비스를

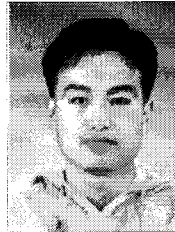
시뮬레이션하는 환경에 적용하여 개발기술에 대한 성능을 검증할 수 있는 것이다.

현재 확보되어 있는 실제 온라인서비스 데이터로부터 불법적 활용과 관련된 패턴에 대한 분석을 마치고, 이를 CEP 엔진에 등록하여 로그 서버로부터 입력되는 데이터로부터 패턴을 실시간으로 탐지할 수 있는 기술적인 부분을 마무리 짓는다. 탐지 결과는 개발 중인 사용자 인터페이스에 적절히 나타낼 수 있는 형태로 제공된다. 온라인서비스의 로그 서버, CEP 엔진, 그리고 사용자 인터페이스가 통합 구현된 상태에서의 성능 평가를 수행하여, 목표한 성능 수준을 만족하는지 확인하고, 세부적인 튜닝을 거쳐 프로토타입 감시 도구를 완성하도록 한다.

저 자 소개

노 창 현(Chang Hyun Roh)

중신회원



- 1993년 2월 : KAIST 원자력공학과(공학사)
  - 1995년 2월 : KAIST 원자력공학과(공학석사)
  - 2001년 2월 : KAIST 원자력공학과(공학박사)
  - 2006년 3월 ~ 2007년 2월 : 엠게임 정보기획실장, 엠게임 USA 개발 이사
  - 2007년 3월 ~ 2007년 6월 : 엔트리소프트 미국지사 자문위원
  - 2002년 3월 ~ 현재 : 중부대학교 게임학과 교수
- <관심분야> : 가상세계, 온라인게임, 게임기획, Interactive Media

참 고 문 헌

- [1] 대한민국 게임백서 2008, 한국게임산업개발원 산업정책팀, 한국게임산업진흥원, 2008.
- [2] 최용준, "온라인게임 서비스 안정화," 전자통신동향분석, 제22권 제4호, pp. 43-44, 2007(8).
- [3] 손성호, "프로세스 기반 이벤트 분석을 이용한 비즈니스 활동 모니터링", 한국전자거래학회지, 제12권, 제2호, pp.219-231, 2007.
- [4] <http://esper.codehaus.org>