

## 유비쿼터스 센서 네트워크를 위한 이벤트 처리 기법에 관한 연구

조양현\*, 박용민\*\*, 김현규\*\*\*

### A study on the event processing methods for ubiquitous sensor network

Yang-Hyun Cho\*, Yong-Min Park\*\*, Hyeon-Gyu Kim\*\*\*

#### 요 약

RFID(Radio Frequency Identification)와 WSN(Wireless Sensor Network)은 기술적 유사성과 상호 영향에도 불구하고 별 개의 연구로 인식되었으며 RFID와 WSN의 기술적인 융합에 대한 연구는 미비한 수준이다. 이러한 문제점을 인식한 EPCglobal에서는 국제 표준인 EPCglobal Network를 기반으로 RFID와 WSN 기술을 효율적으로 통합/연동할 수 있는 EPC(Electronic Produce Code) 센서 네트워크를 제안하였다. 제안된 EPC 센서 네트워크는 미들웨어에서 복합 이벤트 처리(Complex Event Processing) 기법을 이용하여 RFID와 WSN을 통해 발생하는 데이터를 단일 환경에서 통합하고, 발생된 이벤트를 EPC global Network를 기반으로 연동하는 기술이다. 하지만 이러한 EPC 센서 네트워크는 미들웨어에서 복합 이벤트를 검출하기 위해 복합 이벤트를 구성하는 모든 기본 이벤트가 발생하지 못하는 경우에도 계속적인 검사를 수행하기 때문에 연산 비용이 증가되는 문제점이 있다. 이러한 EPC 센서 네트워크의 문제점들을 해결하기 위하여 본 논문에서는 미들웨어에서 복합 이벤트의 최소 조건을 정의하고 조건을 만족하는 경우에만 복합 이벤트를 검출하는 알고리즘을 제안하였다. 복합 이벤트의 최소 조건을 검사하기 위하여 인덱스 기반의 질의 색인을 사용하여 복합 이벤트를 추출한다. 제안하는 인덱스 기반의 복합 이벤트 검출 알고리즘은 기존의 검출 알고리즘과 NS2 시뮬레이션을 이용하여 성능 평가를 수행하여 복합 이벤트를 검출하는 시간에서는 기존 알고리즘과 비교하여 평균 6.8msec 적게 나타났으며, 검출에 필요한 연산수는 기존 알고리즘과 비교하여 평균 226개로 적은 연산 수를 나타내었다. 결과적으로 기존 알고리즘에 비해 복합 이벤트 검출을 위해 제안하는 알고리즘이 우수한 성능을 나타내었다.

▶ Keywords : 유비쿼터스, USN, RFID, WSN, EPC글로벌 네트워크

- 
- 제1저자 : 김현규, 교신저자 : 조양현, 책임저자 : 박용민
  - 투고일 : 2012. 11. 09, 심사일 : 2012. 12. 26, 게재확정일 : 2013. 1. 15.
  - \* 삼육대학교 컴퓨터학부(Dept. of Computer Science & Engineering, Sahmyook University)
  - \*\* 삼육보건대학 의료정보시스템과(Dept. of Medical Information System, Sahmyook College)
  - \*\*\* 삼육대학교 컴퓨터학부(Dept. of Computer Science & Engineering, Sahmyook University)
  - ※ 이 논문은 삼육대학교 학술연구비(RI자율2012066)의 지원 및 관리로 수행되었습니다.

## Abstract

The RFID(Radio Frequency Identification) and the WSN(Wireless Sensor Network) have technical similarities and mutual effects, they have been recognized to be studied separately, and sufficient studies have not been conducted on the technical integration of the RFID and the WSN. Therefore, EPC global which realized the issue proposed the EPC(Electronic Produce Code) Sensor Network to efficiently integrate and interoperate the RFID and WSN technologies based on the international standard EPC global network. The proposed EPC Sensor Network technology uses the Complex Event Processing method in the middleware to integrate data occurring through the RFID and the WSN in a single environment and to interoperate the events based on the EPC global network. However, as the EPC Sensor Network technology continuously performs its operation even in the case that the minimum conditions are not to be met to find complex events in the middleware, its operation cost rises.

Therefore, to address the problems of the existing system, we defined the minimum conditions of the complex events to detect unnecessary complex events in the middleware, and proposed an algorithm that can extract complex events only when the minimum conditions are to be met. To examine the minimum conditions, a index and a query index are used to extract complex events. To evaluate the performance of the proposed methods, In the case of the method of extracting complex events based on a bitmap index, we used the existing extraction method and NS2 simulation to evaluate its performance and thus to show its good performance in terms of the number of operation and the processing time for the complex events.

▶ Keywords : ubiquitous, USN, RFID, WSN, EPCglobal Network

## I. 서론

유비쿼터스 컴퓨팅을 위한 핵심 기술은 RFID와 WSN을 이용하여 실제 물리적 환경 정보를 원격에서 손쉽게 수집하고 모니터링하여 기존 가상 공간의 IT 기술을 다양한 실제 환경으로 확대할 수 있는 기술이다. 이러한 RFID와 WSN은 식별기술이므로 위치를 관리하는 프레임워크와 같이 동작하여 물체의 식별 및 위치추적 시스템을 구축할 수 있다. 기존의 연구에서는 EPC를 기반한 특별한 미들웨어를 사용하는 기법이 소개되었다. 그러나 미래에는 다양한 네트워크 기술들이 인터넷 프로토콜(IP)을 기반으로 수렴하여 이른바 All IP 네트워크를 구성할 것으로 기대된다. 따라서 IP 기반의 USN 관리 시스템은 적은 비용으로 설치가 가능하고 기존의 IP 기반의 네트워크와의 쉬운 연동을 가능하게 할 것이다[1][2].

그러나 RFID와 WSN은 이러한 기술적 유사성과 상호 영향에도 불구하고 별개의 연구로 인식되었으며 RFID와 WSN의 기술적인 융합에 대한 연구는 미비한 수준이다. 또한 지금까지의 WSN은 미리 준비된 환경에서 제한된 사용자만을 위한 특정 어플리케이션을 위한 서비스로 제공되었기 때문에 인터넷을 통한 연동은 제공되지 않았다. 따라서 무선 센서 네트워크를 인터넷으로 확장해서 다양한 형태의 서비스를 제공하지 못했으며 다른 센서 네트워크간의 데이터를 공유할 수 있는 표준화가 마련되지 못했다[3]. 이러한 문제점을 인식한 EPCglobal에서는 Auto-ID Lab에서 RFID의 인터넷 아키텍처에 WSN을 결합한 EPC 센서 네트워크를 제안하였다. EPC 센서 네트워크는 국제 표준인 EPCglobal Network를 기반으로 RFID와 WSN 기술을 인터넷과 통합하는 아키텍처를 제공함으로써 언제 어디서나 인터넷을 기반으로 다양한 센서 네트워크와 RFID 서비스를 제공하는 유비쿼터스 컴퓨팅

환경을 제공하는데 목적을 두고 있다(4). EPC 센서 네트워크는 RFID와 WSN을 통해 발생하는 데이터를 단일 환경에서 통합 처리하기 위해 미들웨어에서 복합 이벤트 처리(Complex Event Processing) 기법을 이용하여 EPC global Network에서 센서 데이터를 처리할 수 있게 된다. 복합 이벤트 처리 방식은 여러 이벤트 소스로부터 발생한 이벤트를 대상으로 이벤트들의 영향을 분석하여 대응되는 액션을 처리하는 방식으로 가장 핵심이 되는 기술은 어떻게 사용자에게 의미있는 이벤트(복합 이벤트)를 검출하는가이다(5)(6). 복합 이벤트는 기본 이벤트의 조합으로 이루어진 이벤트로, 복합 이벤트를 구성하는 모든 이벤트가 발생하여야지만 응용 계층으로 결과를 전달할 수 있다. EPC 센서 네트워크에서 제시하는 복합 이벤트 처리 방식은 아직까지 표준으로 정의되진 않았지만, 여러 논문을 통해 가능성을 제시하였다. 지금까지 제시된 복합 이벤트 검출 알고리즘은 오토마타 기법을 이용한 SASE 알고리즘(7)과 복합 이벤트의 패턴을 그래프 형태로 구성하고 발생된 이벤트를 검출하는 RCEDA 알고리즘(8)이 대표적이다. 이러한 알고리즘은 공통적으로 발생하는 모든 기본 이벤트를 상위노드로 전달하여 비교 연산을 수행하기 때문에 기본 이벤트의 수가 증가함에 따라 복합 이벤트를 검출하기 위한 연산 비용이 크게 증가하는 문제점을 가지고 있다. 본 논문에서는 기존 알고리즘의 문제점들을 해결할 수 있는 미들웨어에서 대용량 RFID와 WSN 데이터 스트림에 대한 효율적인 복합 이벤트 처리를 위해 기존 연구의 문제점을 해결한 새로운 알고리즘을 제안하였다. 제안하는 알고리즘은 복합 이벤트를 질의 형태로 등록하고 기본 이벤트가 발생하면 질의 색인을 통하여 복합 이벤트에 접근한다. 질의는 복합 이벤트를 구성하는 기본 이벤트의 발생 여부만을 비교하고 복합 이벤트를 구성하는 기본 이벤트가 모두 발생 할 때 실제 복합 이벤트 검출 연산을 수행한다. 본 논문의 세부 구성은 다음과 같다. 제 2장에서는 관련연구로 EPC 센서 네트워크 및 이벤트 처리의 문제점을 설명하고, 제3장에서는 제안하는 미들웨어에서의 복합 이벤트 검출 알고리즘에 관하여 설명한다. 제 4장에서 제안하는 알고리즘의 성능 평가를 수행하고 결과를 설명한다. 마지막으로 제5장에서는 결론을 맺는다.

## II. 관련 연구

### 1. ESN(EPC 센서 네트워크)

RFID는 각 사물들의 글로벌한 식별을 가능하게 하고 RFID에서 사실 상의 표준인 EPCglobal Architecture Framework는 RFID 데이터를 처리하기 위한 프레임워크로서 만들어졌다. 하지만 EPCglobal Architecture Framework는 RFID 데이터만을 위해서 디자인된 아키텍처이므로 사물들의 주변 환경 정보와 같은 데이터를 제공하는 WSN에 대해서는 고려하지 않았다. 반대로 WSN의 경우 다양한 센서 네트워크가 존재하면서 하나의 통일된 규격이 존재하지 않는다. 이를 보완하기 위해 RFID와 WSN를 통합한 미들웨어 프레임워크인 EPC 센서 네트워크가 제안되었다. EPC 센서 네트워크는 RFID의 인터넷 아키텍처에 WSN를 결합한 인프라로 정의 될 수 있으며 인터넷 연동 기술로써 EPCglobal의 아키텍처 프레임워크를 기반으로 하고 있다. EPC 센서 네트워크에서는 논리적 컴포넌트의 기능을 정의하는 대신에 각각의 통신 인터페이스를 정의하고 있으며 이들 인터페이스를 변경함으로써 RFID 뿐만 아니라 WSN의 센싱 특성을 통합한 다양한 형태의 RFID가 지원될 수 있도록 하였다. 그림 1은 EPC 센서 네트워크의 구조를 보여준다(9-11).

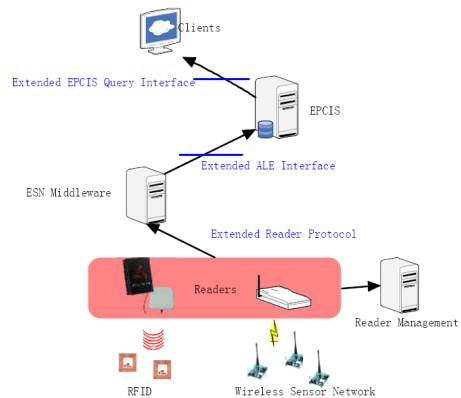


그림 1. EPC 센서 네트워크 구조  
Fig. 1. The structure of EPC sensor network

EPC 센서 네트워크는 센서 데이터의 표현을 위해 EPC를 기반으로 "EPC + 센서 데이터"의 결합된 쌍으로 표현된다. 또한 그림 2와 같이 EPC 센서 네트워크 미들웨어는 이벤트

기반의 복합 이벤트 처리(Complex Event Processing)를 이용한 이벤트 처리 컴포넌트로 EPCglobal Network에서 센서 데이터의 처리에 사용된다. EPC 센서 네트워크에서의 복합 이벤트 처리 연구는 기존 EPCglobal Network의 ALE 미들웨어 외에 자체적인 미들웨어 구조로 디자인하였다. 복합 이벤트 처리는 여러 이벤트 소스로부터 발생한 이벤트를 대상으로 이벤트들의 영향을 분석하여 대응되는 액션을 처리하는 방식으로 가장 핵심이 되는 기술은 어떻게 사용자에게 의미있는 이벤트(복합 이벤트)를 검출하는가이다. EPC 센서 네트워크에서 제시하는 이벤트 검출 알고리즘은 그래프를 이용하여 USN 데이터를 위한 그래프 기반의 복합 이벤트를 처리하는 알고리즘을 제안하였다[12][13].

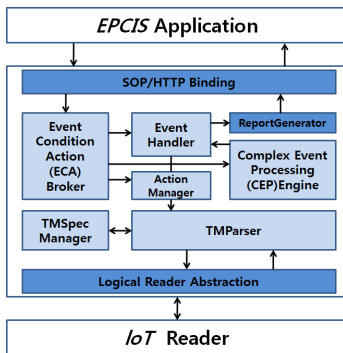


그림 2. EPC 센서 네트워크 미들웨어  
Fig. 2. Middleware of EPC sensor network

2. 그래프 기반의 복합 이벤트 처리

그래프 기반의 복합 이벤트 처리에서는 복합 이벤트를 효율적으로 처리하기 위하여 그림 3과 같이 복합 이벤트를 정의하는 규칙을 제안하였고 복합 이벤트를 구성하기 위해 기본 이벤트를 조합하기 위한 연산자를 정의하였다. 이러한 연산자는 그림 4와 같이 AND, OR, NOT과 같은 기본 연산자와 순차적인 이벤트를 묶는 SEQ, 시간 조건이 포함된 TSEQ 연산, 동일한 이벤트의 연속적인 처리를 위한 SEQ+, TSEQ+ 연산 등이 있다.

```
CREATE RULE rule_id, rule name
ON event
IF condition
DO action1; action2; ...; actionn
```

그림 3. 복합 이벤트 규칙  
Fig 3. Complex event rule

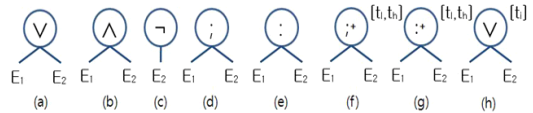


그림 4. 복합 이벤트를 구성하는 기본 연산자 그래프  
Fig. 4. The basic operator graph to organize complex events

실제 복합 이벤트를 처리하기 위해 어플리케이션에서 등록된 복합 이벤트는 계층적인 그래프로 저장한다. 그래프의 단말노드는 단순 이벤트로 구성되어 있으며 비 단말노드는 이벤트의 연산자로 구성되어 있다. 단순 이벤트가 발생하면 그래프의 단말 노드에서부터 이벤트의 조건을 검사하여 상위 노드로 전달된다. 이때, 발생하지 않는 이벤트의 연산자가 포함된 노드를 만난 경우는 상위 노드에서 Pseudo 이벤트를 발생시킨다. 발생하지 않는 연산은 하위 노드에서 상위 노드로 이벤트를 전달하는 시점을 선정할 수 없기 때문에 발생한 Pseudo 이벤트는 정해진 시간에 하위 노드로 질의를 전달하고 조건을 만족하는 경우 다시 상위 노드로 전달된다. 이벤트가 최종 루트 노드에서 조건을 만족하게 되면 복합 이벤트를 등록된 응용 계층에게 최종 이벤트가 전달된다[14][15].

3. 그래프 기반의 복합 이벤트 처리의 문제점

EPC 센서 네트워크는 미들웨어에서 USN 데이터를 통합 처리하기 위해 복합 이벤트 처리 방식을 제안하였다. 복합 이벤트 처리 방식은 여러 이벤트 소스로부터 발생한 이벤트를 대상으로 이벤트들의 영향을 분석하여 대응되는 액션을 처리하는 방식으로 가장 핵심이 되는 기술은 어떻게 사용자에게 의미있는 이벤트(복합 이벤트)를 검출하는가이다. 복합 이벤트는 기본 이벤트의 조합으로 이루어지기 때문에 복합 이벤트가 발생하기 위해서는 복합 이벤트를 구성하는 기본 이벤트가 모두 발생해야 한다. 본 논문에서는 복합 이벤트를 구성하는 모든 기본 이벤트가 모두 발생한 시점을 복합 이벤트의 최소 조건을 만족한다고 한다. EPC 센서 네트워크에서 제시한 복합 이벤트 검출 알고리즘은 복합 이벤트를 그래프의 형태로 등록하고 발생하는 모든 기본 이벤트는 그래프의 상위 노드로 전달되어 복합 이벤트 검출을 수행하였다. 이는 발생하는 기본 이벤트의 수가 증가함에 따라 연산량이 크게 증가되는 단점이 있다. 또한, 복합 이벤트의 최소 조건을 만족하지 못하는 상황에서도 복합 이벤트를 추출하기 위한 연산을 수행한다. 이때, 복합 이벤트 검출 연산은 완벽한 복합 이벤트를 생성하지 못하고 중간 단계의 후보 집합만을 연산할 수 있다. 중간 단계의 후보 집합은 복합 이벤트를 구성하는 다른 기본 이벤트의 발생 유무에 의해 최종 복합 이벤트의 후보 집합으

로 생성하거나 삭제한다. 삭제되는 중간 단계의 후보 집합은 불필요한 연산을 의미하기 때문에 불필요한 중간 단계의 후보 집합을 만드는 과정은 시스템의 자원을 낭비한다. 그림 5는 기존 방식의 문제점을 나타낸 것이다. E1, E2, E3, E4의 이벤트로 구성된 복합 이벤트가 있고 기본 이벤트의 E1, E3, E4는 빈번히 발생하지만 E2 이벤트는 발생하지 않는다. 이러한 경우에서 중간 단계의 후보 집합인 E3, E4의 조합 결과는 계속 생성되지만 최종 복합 이벤트는 발생하지 않는다.

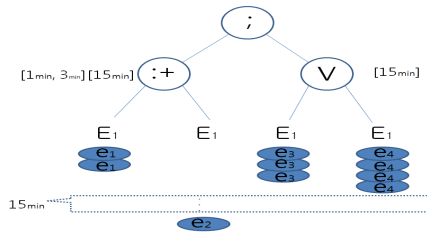


그림 5. 기존 알고리즘의 이벤트 처리 문제점  
Fig. 5. The event processing problems traditional methods

기존 알고리즘에서 기본 이벤트가 발생하면 하위 노드에서 상위 노드로 전달된다. 이벤트를 전달 받은 상위 노드는 하위 노드에서 발생한 이벤트를 저장하고 하위 노드들로부터 전달 받은 이벤트를 이용하여 중간 단계의 후보 집합을 생성하여 상위 노드로 전달해야 한다. 따라서 상위 노드는 모든 중간 단계의 후보 집합 이벤트를 저장하기 위한 추가적인 구조가 필요하다. 실제 복합 이벤트의 발생에 영향을 주지 않는 불필요한 중간 단계 후보 집합도 상위 노드로 전달되어 저장하기 때문에 불필요한 저장 비용을 소모한다. 불필요한 중간 단계의 후보 집합은 상위 노드에서 또다시 불필요한 중간 단계의 후보 집합을 생성한다. 이는 복합 이벤트 추출 연산에서 다시 비교 연산을 수행하는 대상이 되기 때문에 복합 이벤트 검출 시간을 증가시키는 원인이 된다. 기존 알고리즘에서는 여러 복합 이벤트들의 그래프 공유에 대한 문제점이 발생한다. 복합 이벤트는 서로 다른 복합 이벤트와 그래프의 노드를 공유할 수 있다. 하지만 연산을 구성하는 연산자와 하위 노드의 연산 시간이 모두 일치하지 않으면 공유할 수 없다. 따라서 그래프의 단말 노드들은 공유가 용이하지만 비 단말 노드는 쉽게 공유되지 않는다.

### III. 제안하는 이벤트 검출 알고리즘

본 장에서는 제안하는 이벤트 검출 알고리즘을 위한 시스템 구성도, 인덱스, 이벤트 질의, 이벤트 검출 과정에 대해 기

술한다. 기존 연구의 문제점을 해결하기 위해 제안하는 복합 이벤트 검출 알고리즘은 복합 이벤트를 구성하는 단순 이벤트들이 모두 발생할 때 복합 이벤트를 검출하도록 검출 시점을 조절한다. 제안하는 검출 알고리즘은 그림 6과 같이 복합 이벤트를 구성하는 모든 단순 이벤트들이 발생되었는지를 판별하기 위해 인덱스를 구성하며, 복합 이벤트를 위한 질의는 복합 이벤트를 구성하는 기본 이벤트중 마지막 이벤트를 트리거 이벤트로 정의하고, 트리거 이벤트를 기준으로 복합 이벤트 질의를 등록한다. 복합 이벤트 질의는 연산자의 특성에 의해 트리 형태로 구성된다. 실제 단순 이벤트가 발생하면 인덱스를 통해 발생시간을 기록하며, 만약 복합 이벤트를 구성하는 모든 단순 이벤트가 발생하면 등록된 복합 이벤트를 검출하기 위해 복합 이벤트 트리를 검사한다.

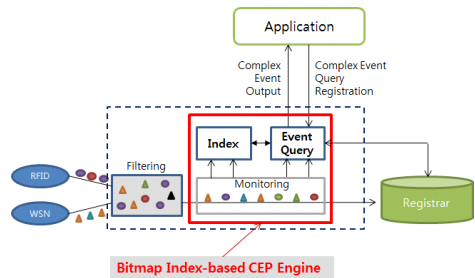


그림 6. 복합 이벤트 처리를 위한 모델  
Fig. 6. The proposed model for complex event processing

#### 1. 이벤트 질의

이벤트 질의는 응용 계층에서 등록된 복합 이벤트를 효율적으로 처리하기 위하여 복합 이벤트 정보를 저장하는 구조이다. 이벤트 질의는 응용 계층에서 등록된 복합 이벤트를 연산자의 특성에 의해 트리 형태로 구성한다. 또한 등록된 복합 이벤트를 구성하는 모든 이벤트 중 마지막 이벤트를 트리거 이벤트로 정의하며, 이 트리거 이벤트를 기준으로 복합 이벤트 질의를 등록하게 된다. 트리거 이벤트는 정의 1은 다음과 같다.

[정의 1] 트리거 이벤트
트리거 이벤트는 복합 이벤트를 추출할 수 있는 기준 시점을 제공하는 이벤트로 연산자에 따라 정의되며 SEQUENCE 연산의 경우 마지막에 위치한 이벤트를 트리거 이벤트로 정의하며, 다른 연산자에 의한 연결된 이벤트들은 마지막 항에 연결된 모든 이벤트를 트리거 이벤트로 정의한다.

이벤트를 트리거 이벤트로 정의하며, 다른 연산자에 의한 연결된 이벤트들은 마지막 항에 연결된 모든 이벤트를 트리거 이벤트로 정의한다. 예를 들어, 그림 7과 같이 복합 이벤트를 위한 질의 Q1, Q2, Q3가 응용 계층에 등록되었다면 이벤트 질의는 트리 형태로 구성되며, 트리거 이벤트를 기준으로 복합 이벤트 질의를 등록하게 된다. 여기서 트리거 이벤트는 d, c, h가 된다. 또한 하나의 트리거 이벤트(d)는 다른 복합 이벤트들 사이에서 공유될 수 있다.

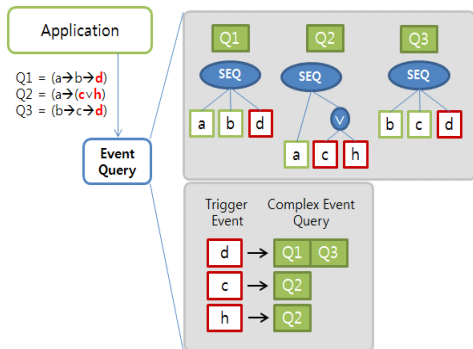


그림 7. 이벤트 질의 구조  
Fig. 7. Structure of event query

2. 검출 과정

그림 8은 복합 이벤트 검출을 위한 처리 과정을 나타낸다. 먼저 응용 계층으로부터 수신된 복합 이벤트는 이벤트 질의를 통해 트리 형태와 트리거 이벤트를 기준으로 복합 이벤트를 등록한다. 수신된 기본 이벤트에서 트리거 이벤트인 d의 기본 이벤트가 발생하였다면 인덱스에 발생한 시간을 기록하며, 이벤트 질의를 통해 자신을 포함한 복합 이벤트인 Q1과 Q3를 검색한다. Q1과 Q3 각각의 복합 이벤트는 d 이벤트를 제외한 다른 이벤트의 발생 유·무를 식별하기 위해 인덱스를 통해 검색한다. 인덱스의 검색은 복합 이벤트 질의가 최초 등록된 시점에서 트리거 이벤트 및 복합 이벤트를 구성하는 기본 이벤트가 발생한 시점까지를 시간 범위로 하여 그 범위 내에서 각 질의를 구성하는 이벤트의 발생 유·무를 검색한다. 만약 모든 이벤트가 발생하였다면 각각의 복합 이벤트는 연산을 수행하며 그렇지 못하다면 연산을 수행하지 않는다. Q1과 Q3는 SEQUENCE 연산자로 발생 순서가 중요하다. 따라서 Q1에서 d 이벤트가 T4에서 발생하였다면, 그 다음 순서인 b 이벤트는 T0~T4 시간 범위에서 발생 유·무를 검색한다. b 이벤트는 T3 시점에서 발생하였으므로, a 이벤트를 b 이벤트의 발생 시점을 기준으로 T0~T3 시간범위에서 다시 발생 유·무를 검색한다. 그 범위의 T2에서 발생하였기 때문에 Q1

질의는 모든 이벤트가 발생하였으므로 복합 이벤트 검출을 위한 연산을 수행한다. Q3의 경우 마찬가지로 d 이벤트가 T4에서 발생하였다면 c 이벤트는 T0~T4 시간범위에서 발생 유·무를 검색한다. c 이벤트는 그 시간범위에서 발생하지 않았으므로 복합 이벤트 검출을 위한 연산을 수행하지 않는다.

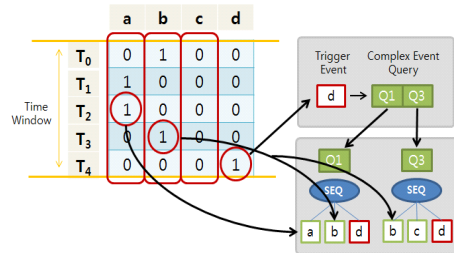


그림 8. 복합 이벤트 검출을 위한 처리 과정  
Fig. 8. The processing for complex event detection

3. 제안하는 알고리즘의 평균 처리 시간

제안하는 복합 이벤트 검출을 위한 평균 처리 시간은 등록되는 복합 이벤트 수와 기본 이벤트의 발생으로 복합 이벤트 처리 엔진에서 처리 시간을 측정한다. 식 1은 등록되는 복합 이벤트 수(C)를 나타낸 것으로 Min은 등록되는 복합 이벤트의 초기값을 나타내며, Max는 등록되는 복합 이벤트의 최대값을 나타낸다. rand()는 복합 이벤트의 초기값과 최대값의 범위에서 무작위로 수를 추출한다. 또한 식 2는 각 복합 이벤트를 구성하는 기본 이벤트 수(PE)를 나타낸 것으로 복합 이벤트 수와 같이 일정한 범위에서 무작위로 추출할 수 있도록 정의 된다. P\_Max는 각 복합 이벤트를 구성하는 기본 이벤트의 최대값을 나타내며 P\_Min은 각 복합 이벤트를 구성하는 기본 이벤트의 초기값을 나타낸다.

$$C = (rand() \% (Max - Min) + Min) \tag{1}$$

$$PE = (rand() \% (P_{Max} - P_{Min}) + P_{Min}) \tag{2}$$

제안하는 복합 이벤트 검출 알고리즘은 인덱스를 통해 복합 이벤트를 구성하는 모든 기본 이벤트의 발생 유·무를 관리한다. 따라서 평균 처리 시간은 인덱스를 검색하는 시간을 포함한다. 식 3은 인덱스를 검색하는데 걸리는 시간(T\_search)을 나타낸다. K = 1인 경우는 복합 이벤트를 구성하는 모든 기본 이벤트가 발생한 경우를 나타내며 K = 0인 경우는 복합 이벤트를 구성하는 모든 기본 이벤트가 발생하지 않는 경우를 나타낸다. B\_time[k]는 인덱스에서 k 번째 기본 이벤

트를 검색하는데 걸리는 시간을 나타내며, 1에서 PE의 범위에서 발생한 기본 이벤트의 검색 시간의 합이 인덱스를 검색하는 전체 시간이 된다.

$$T_{search} = \begin{cases} K=1, & \sum_{1 \leq k \leq PE} B_{time}[k] \\ K=0, & 0 \end{cases} \quad (3)$$

1개의 복합 이벤트를 처리하는 시간( $C_{time}$ )은 식 4와 같이 인덱스를 검색하는 시간( $T_{search}$ )과 1개의 복합 이벤트를 구성하는 모든 기본 이벤트가 발생하였을 경우 기본 이벤트를 처리하는 시간( $P_{time}[k]$ )의 합으로 나타낼 수 있다. 기존 알고리즘과의 차이점은 인덱스를 검색하는 걸리는 시간과 복합 이벤트를 구성하는 모든 기본 이벤트가 발생하는 경우만 복합 이벤트 검출 연산을 수행하는 것에 차이점이 있다.

$$C_{time} = T_{search} + \begin{cases} K=1, & \sum_{1 \leq k \leq PE} P_{time}[k] \\ K=0, & 0 \end{cases} \quad (4)$$

전체 등록된 복합 이벤트를 검출하는데 걸리는 평균 처리 시간( $C_{totaltime}$ )은 식 4에 의해 전체 등록된 복합 이벤트를 처리하는데 걸리는 시간을 등록된 복합 이벤트 수로 나누어서 식 5와 같이 계산 할 수 있다. 여기서  $Min$ 은 등록된 복합 이벤트의 최소값이며  $C$ 는 등록된 복합 이벤트 수를 나타낸다.

$$C_{totaltime} = \frac{\sum_{Min \leq i \leq C} C_{time}[i]}{C} \quad (5)$$

#### 4. 제안하는 알고리즘의 이벤트 검출을 위한 연산 수

연산 수는 등록된 복합 이벤트를 검출하는데 발생한 연산의 수를 나타내며, 각 복합 이벤트를 구성하는 연산자 수에 따라 발생한 연산 수를 나타낼 수 있다. 식 6은 각 복합 이벤트를 구성하는 연산자 수(POnum)를 나타내는 식으로  $PO_{Max}$ 는 각 복합 이벤트를 구성하는 연산자의 최대값을 나타내며,  $PO_{Min}$ 는 각 복합 이벤트를 구성하는 연산자의 초기값을 나타낸다.  $rand()$ 는 각 복합 이벤트를 구성하는 연산자의 초기값과 최대값의 범위에서 무작위로 수를 추출한다. 따라서 모든 복합 이벤트 검출을 위해 사용되는 연산 수

( $PO_{totalnum}$ )는 식 7과 같다.

$$PO_{num} = (rand())\%(PO_{Max} - PO_{Min}) + PO_{Min} \quad (6)$$

$$PO_{totalnum} = \begin{cases} K=1, & \sum_{1 \leq k \leq C} PO_{num}[k] \\ K=0, & 0 \end{cases} \quad (7)$$

## IV. 성능평가 및 고찰

### 1. 실험 환경

본 장에서는 제안하는 이벤트 처리 알고리즘의 성능을 기존 알고리즘과 비교 분석한다. 미들웨어에서의 복합 이벤트 검출 성능 평가에 사용된 시스템은 Intel Dual Core 2.4GHz 프로세서와 2Gbyte의 메모리를 가지고 있으며 운영체제는 Fedora Linux 9를 사용하였다. 복합 이벤트 처리를 위한 모듈은 NS2 시뮬레이터를 이용하여 구현하였다. 성능 평가는 응용 계층으로부터 등록되는 복합 이벤트의 수와 초당 발생하는 기본 이벤트의 수에 따라 기존 연구에서 복합 이벤트 검출을 위한 평균 처리 시간과 제안하는 알고리즘에서의 평균 처리 시간을 비교한다. 또한 기본 이벤트 수를 변경하여 복합 이벤트 검출을 위한 연산 수를 비교 분석한다. 그림 9는 미들웨어에서 복합 이벤트 처리를 위한 성능 평가 모델을 나타낸다. 성능 평가 모델은 복합 이벤트 발생기를 통해 복합 이벤트를 복합 이벤트 처리 엔진에 등록하며, 기본 이벤트 발생기는 RFID와 WSN 통해 발생하는 기본 이벤트를 나타낸다. 복합 이벤트 처리 엔진에서는 등록된 복합 이벤트를 기반으로 발생하는 기본 이벤트를 검출하는 기능을 수행한다.

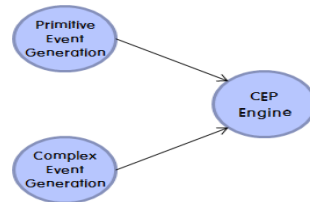


그림 9. 성능 평가 모델  
Fig. 9. Performance evaluation model

표 1은 복합 이벤트 검출을 위해 기본 이벤트와 복합 이벤트 발생을 위한 파라미터를 나타낸다. 기본 이벤트 발생기는 초당 10,000~50,000개를 발생할 수 있도록 설계 하였으며, 이벤트 형식은 RFID는 (ID, L, T)로 구성되며, WSN은

(ID, L, D, T)로 구성된다. ID는 각 태그 및 센서 노드의 식별자를 나타내며, L은 센싱 위치, T는 센싱 시간을 나타내며, WSN의 D는 센싱 데이터를 나타낸다. 또한 복합 이벤트 발생하는 복합 이벤트 등록을 100~500개로 하였으며, 복합 이벤트의 순서 길이는 SEQUENCE 연산자를 통해 이벤트의 발생 순서를 정의하는 것으로, 2~10의 길이를 가질 수 있도록 설계하였다. 또한 각 복합 이벤트의 기본 이벤트 수는 2~10개의 기본 이벤트 수를 가질 수 있도록 설계하였으며, 연산자 수는 0~20개의 연산자를 가질 수 있도록 설계하였다. 복합 이벤트는 트리 형태로 구성되며, 트리는 2진 트리 구성이 된다. 트리의 높이를 1~3에 따라 성능 평가를 수행할 수 있도록 설계하였다.

표 1. 성능평가를 위한 파라미터

Table 1. Parameter for performance evaluation

Parameters for primitive event generation		
Parameter	Description	Values
P	Primitive events occur per second (event/sec)	[10,000-50,000]
T	Number of primitive event types (ID, L, D, T)	[1-100]
$\theta_1$	Random number generate of occurrences of event types	[1-100]

Parameters for complex event generation

Parameter	Description	Values
C	Number of complex events registered	[100-500]
S	Length of the sequence in each complex event	[2-10]
PE	Number of primitive event in each complex event	[1-20]
O	Number of operator in each complex event	[0-20]
L	Number of tree level in each complex event	[1-3]
$\theta_2$	Random number generate of PE, O, L	

2. 실험 결과

이 절에서는 복합 이벤트의 트리 레벨에 따른 성능 평가를 수행하기 위해 트리의 레벨을 변경하면서 복합 이벤트 처리 시간과 연산의 수를 평가한다. 실험을 위한 복합 이벤트는 2진 트리로 구성되며, 복합 이벤트는 500개로 등록하여 실험을 수행하였다.

(1) 트리의 레벨을 1로 설정

복합 이벤트의 트리 레벨에 따른 연산 시간을 비교하기 위해 트리의 레벨을 1로 설정하고 초당 발생하는 복합 이벤트의 수를 변경하면서 이벤트를 처리하는 평균 시간을 측정하였다. 그림 10은 복합 이벤트를 처리하는데 소요되는 평균 처리 시간을 나타낸다. 제안하는 알고리즘과 기존 알고리즘 모두 기본 이벤트가 증가함에 따라 선형적으로 연산 시간이 증가함을 볼 수 있다. 이때, 각각 증가하는 비율은 거의 유사

하다. 트리의 레벨이 1인 경우 제안하는 알고리즘이 기존 알고리즘과 비교하여 평균 3.9msec의 연산 시간이 더 소요됨을 볼 수 있다. 이는 제안하는 알고리즘이 기존 알고리즘보다 복합 이벤트를 구성하는 모든 기본 이벤트의 발생 유·무를 검사하기 위한 검색 시간을 추가적으로 수행하기 때문이다. 또한, 복합 이벤트의 트리 레벨이 작을수록 복합 이벤트를 구성하는 모든 기본 이벤트의 검출이 간단하기 때문에 연산 시간이 적어진다.

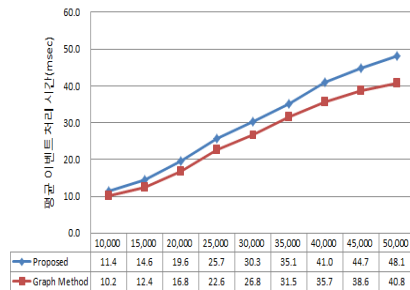


그림 10. 트리 레벨 1에서 처리 시간  
Fig. 10. Processing time at tree level 1

그림 11은 기본 이벤트의 변화에 따라 기존 알고리즘에서 처리하는 복합 이벤트 검출을 위한 연산 수와 제안하는 알고리즘에서 복합 이벤트 검출을 위한 연산 수를 나타낸다. 이 실험에서 제안하는 알고리즘이 복합 이벤트 검출을 위한 연산 수가 평균 70개 가량 적음을 볼 수 있었다. 이는 복합 이벤트 검출의 경우 복합 이벤트를 구성하는 모든 기본 이벤트가 발생할 경우만 연산을 수행하기 때문이다.

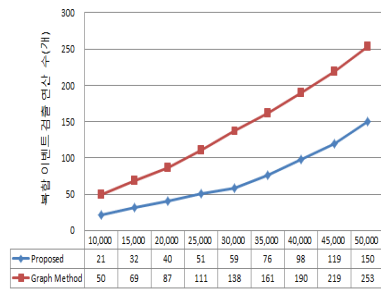


그림 11. 트리 레벨 1에서 연산 수  
Fig. 11. The number of operands at tree level 1

(2) 트리의 레벨을 2로 설정

트리의 레벨을 2로 설정하여 위와 같은 실험을 반복하였다. 그림 12는 트리의 레벨을 2로 하여 실험을 수행한 결과이



다. 트리 레벨 2에서는 제안하는 알고리즘과 기존 알고리즘이 복합 이벤트 검출을 위한 연산 시간이 거의 비슷함을 알 수 있다. 그림 13은 트리의 레벨이 2일 때 기본 이벤트의 변화에 따라 기존 알고리즘에서 처리하는 복합 이벤트 검출 연산 수와 제안하는 알고리즘에서 복합 이벤트의 검출 연산 수를 나타낸다. 기본 이벤트가 증가함에 따라 기존 알고리즘에서 처리하는 복합 이벤트의 수는 크게 증가하는 반면 제안하는 알고리즘은 서서히 증가되는 것을 알 수 있다.

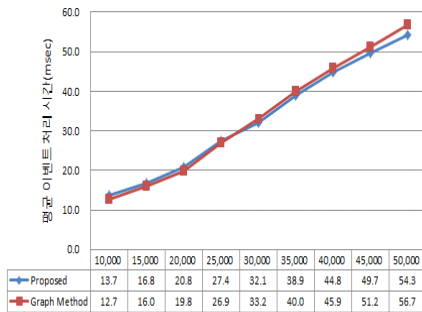


그림 12. 트리 레벨 2에서 처리 시간  
Fig. 12. Processing time at tree level 2

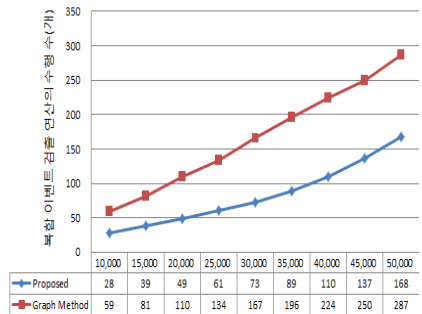


그림 13. 트리 레벨 2에서 연산 수  
Fig. 13. The number of operands at tree level 2

(3) 처리되는 이벤트 수

이 절에서는 초당 30,000개의 기본 이벤트가 발생하는 상황에서 복합 이벤트의 수를 100~500개로 변경시켜 등록하고 검출 연산을 수행하는 연산 수를 측정한다. 이때 복합 이벤트의 트리 레벨은 1~3까지의 임의의 값으로 설정된다. 또한 복합 이벤트를 500개로 고정된 후 기본 이벤트를 10,000~50,000개로 변경 시키며 실험을 반복 수행한다.

그림 14는 복합 이벤트의 변화에 따라 실험 평가한 결과이다. 초당 30,000개의 기본 이벤트가 발생할 때 제안하는 알

고리즘은 기존에 비하여 복합이벤트 검출에 필요한 연산 수가 적음을 알 수 있다. 즉, 이벤트의 복잡도가 다양하면 본 논문에서 제안하는 알고리즘이 처리하는 복합 이벤트 검출 연산의 수가 현저히 감소함을 볼 수 있다.

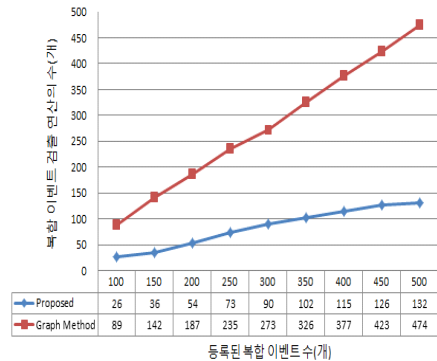


그림 14. 복합 이벤트 수에 따른 연산 수  
Fig. 14. Operations according to the number of complex event

V. 결론

유비쿼터스 환경에서의 가장 핵심이 되는 RFID와 WSN을 이용한 많은 연구가 진행되고 있다. 하지만 RFID와 WSN은 그 기술적인 발달에도 불구하고 표준의 부재와 현재의 기술적 한계 때문에 데이터의 공유와 융합을 위해 이를 지원할 수 있는 인프라스트럭처가 필요하게 되었다. 본 논문에서는 RFID와 WSN 데이터를 기반으로 한 통합 관리 시스템을 제안하였다. 제안하는 관리 시스템은 대용량 RFID와 WSN 데이터를 단일 환경에서 처리하기 위해 미들웨어에서 인덱스 기반의 복합 이벤트 검출 알고리즘을 제안하였다. 제안하는 검출 알고리즘은 복합 이벤트를 구성하는 기본 이벤트의 발생 유·무 판별하기 위해 인덱스로 관리하고 복합 이벤트를 구성하는 기본 이벤트를 이벤트 질의를 통해 트리 구조로 등록하여 관리한다. 복합 이벤트를 구성하는 모든 기본 이벤트가 인덱스를 통해 발생하면 이벤트 질의를 통해 복합 이벤트 검출을 위한 연산을 수행한다. 제안하는 복합 이벤트 검출 알고리즘은 모든 기본 이벤트가 발생할 때만 검출 연산을 수행하기 때문에 연산의 수를 감소시킬 수 있고 불필요한 후보 집합을 생성하지 않아 복합 이벤트 검출을 위한 처리 시간이 줄어들게 된다. 실험을 통해 제안하는 알고리즘이 기존의 EPC 센서 네트워크에서 제안하는 검출 알고리즘에 비해 초당 발생하는 기본 이벤트 수와 복합 이벤트 등록 수에 따라

평균 처리 시간은 6.8msec, 검출에 필요한 연산 수는 평균 226개로 기존 알고리즘에 비해 적게 나타나 우수한 성능을 나타내었다. 향후 과제로는 제안한 알고리즘을 바탕으로 실제 환경에서 실측을 통해 성능의 우수성을 검증할 예정이다.

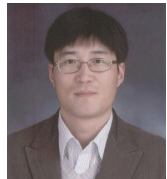
## 참고문헌

- [1] D. Park et. al., "Current Status of Applying RFID/USA Technologies in Medical Services," *Journal of communications and networks, KNOM Review*, Vol 25(10), pp. 50-57, 2008. 9.
- [2] Bang, M, Larsson, A, "Design Requirements for Ubiquitous Computing Environments for Data Professionals," *Studies in health technology and informatics*, pp. 1416-1420, February 2005.
- [3] Y. B. Kim, "u-Data Service Based on a USN Middleware Platform," INC, IMS and IDC, NCM '09. Fifth International Joint Conference on, pp. 673-678, August. 2009.
- [4] J. Sung and D. Kim, "EPC Sensor Network for RFID/USN integrated infrastructure," *KICS Winter Conference*, Vol 23(12), pp. 27-46, 2007.
- [5] K. Lee et. al., "A Middleware for Data Processing and Information Sharing in EPC Sensor Network," *Journal of KISSE*, Vol 37(2), pp.19-22, 2010.
- [6] S. Jung et. al., "A Design of ZigBee Readers for EPC Sensor Network," *Journal of KISSE*, Vol 37(2), pp.28-31, 2010.
- [7] D. Gyllstrom, E. Wu, H. J. Chae, Y. Diao, P. Stahlberg, and G. Anderson SASE: Complex Event Processing over Streams. In *Proceedings of the Third Biennial Conference on Innovative Data Systems Research (CIDR 2007)*, pp.1-3, 2007.
- [8] F. Wang, S. Liu, and Y. Bai, "Bridging Physical and Virtual Worlds: Complex Event Processing for RFID Data stream", *Proc International Conference on Extending Database Technology*, 2006.
- [9] S. Park, "Design and Implementation of a Framework for Developing Effective Applications over EPC Sensor Network," 2009 KIIE Conference, pp. 14-16, 2009.
- [10] Weixin Wang:, Jongwoo Sung, Daeyoung Kim, "Complex Event Processing in EPC Sensor Network Middleware for Both RFID & WSN," *Object Oriented Real-Time Distributed Computing*, 2008 11th IEEE International Symposium on, pp. 165-169, May 2008.
- [11] Sanchez Lopez, T., D. Kim, "A Context Middleware Based on Sensor and RFID Information," *Pervasive Computing and Communications Workshops*, 2007. PerCom Workshops '07. Fifth Annual IEEE International Conference on, pp. 331-336, March. 2007.
- [12] Guangqian Zhang, Li Zhang, "Study of CEP-Based RFID Data Processing Model," *Computer society IEEE*, pp. 185-193, Feb. 2008.
- [13] Louis Perrochon, "Enlisting Event Patterns for Cyber Battlefield Awareness," *DARPA Information Survivability Conference & Expositoin*, Vol. 2, pp.1411-1420, Feb. 2000.
- [14] Hwang, Y. Park, G. Ahn, E. Rho, J. Sung, J. Kim, D. "The Situation Dependent Application Areas of EPC Sensor Network," *Lecture Notes in Computer Science, SPRINGER-VERLAG*, pp. 437-446, 2007.
- [15] Weixin. Wang, "Complex Event Processing in EPC Sensor Network Middleware for Both RFID and Wireless Sensor Networks," *ICU, Thesis(M.A)*, pp.5-7, 2008.

**저 자 소 개**



**조 양 현**  
 1982 : 광운대학교 전자통신공학과  
 공학사.  
 1985 : 광운대학교 전자통신공학과  
 공학석사.  
 2011 : 광운대학교 전자통신공학과  
 공학박사  
 현 재 : 삼육대학교 컴퓨터학부 교수  
 관심분야 : 컴퓨터네트워크, 통신망  
 (BeN), GMPLS  
 Email : yhcho@syu.ac.kr



**박 용 민**  
 2005: 광운대학교  
 전자통신공학과 공학석사  
 2011: 광운대학교  
 전자통신공학과 공학박사  
 현 재: 삼육보건대학교  
 의료정보시스템과 조교수  
 관심분야: 의료정보, 정보통신  
 Email : pym@shu.ac.kr



**김 현 규**  
 1997 : 울산대학교 전산학과(학사)  
 2000 : 울산대학교 전산학과(석사)  
 2000 : 한국국방연구원 연구원  
 2001~05 : LG전자 선임연구원  
 2010 : KAIST 전산학과(박사)  
 2011~2012 : 한국원자력연구원  
 선임연구원  
 현 재 : 삼육대학교 조교수  
 관심분야 : Database,  
 IT Convergence 등  
 Email : hgkim@syu.ac.kr