

# 전처리 필터링을 적용한 효율적인 RFID 비즈니스 이벤트 검출 기법

(An Efficient RFID Business Event Detection Method Using  
Preprocessing Filtering Scheme)

노진석<sup>†</sup>      북경수<sup>\*\*</sup>      유재수<sup>\*\*\*</sup>  
(Rho Jin Seok)    (Kyoung Soo Bok)    (Jae Soo Yoo)

**요약** RFID 이벤트는 연속적으로 발생하는 대용량의 스트림 데이터이다. 연속적인 RFID 스트림으로부터 비즈니스 이벤트를 검출하기 위한 많은 연구들이 진행되었다. 그러나 기존 연구는 실제 비즈니스 이벤트가 발생하는 최소 조건을 만족하지 못하는 경우에도 계속적인 검사를 수행하기 때문에 연산 비용이 증가하는 문제점이 있다. 본 논문에서는 불필요한 비즈니스 이벤트 검출 연산을 제거하기 위하여 비즈니스 이벤트의 최소 조건을 정의하고 조건을 만족하는 경우에만 비즈니스 이벤트를 검출하는 연산을 제안한다. 비즈니스 이벤트의 최소 조건을 검사하기 위하여 비즈니스 이벤트를 그리드 색인에 등록하고 질의 색인과 비트맵을 사용하여 비즈니스 이벤트를 추출한다. 제안하는 비즈니스 이벤트 검출 기법의 우수성을 입증하기 위해서 기존의 연구와 성능 비교 평가를 수행한다.

**키워드** : RFID, RFID 미들웨어, 비즈니스 이벤트, 질의 색인, 이벤트 검출

**Abstract** RFID events are large volume of stream data which come out continuously. Many studies have been done to detect a business event in RFID stream. However, the existing methods have many problems which increase unnecessary operations when business events do not satisfy minimum conditions. In this paper, to remove unnecessary operations, we define the minimum condition of business events and propose an efficient method that detects business events only when the minimum condition is satisfied. To check the minimum condition of business events, we register business queries in a query index. We detect business events using the query index and bitmap. It is shown through various experiment that the proposed method outperforms the existing methods.

**Key words** : RFID, RFID Middleware, Business Event, Query Index, Event Detection

## 1. 서론

RFID(Radio Frequency IDentification)는 유비쿼터스 사회의 핵심 기술로서 모든 사물에 전자태그를 부착하고 무선통신 기술을 이용하여 사물의 정보 및 주변 상황 정보를 감지하는 인식기술로 활용되고 있다[1,2]. RFID 시스템은 RFID 태그, RFID 리더와 안테나, 미들웨어, 응용 서비스로 구성된다. 태그는 객체의 식별자를 포함하고 있으며 RFID 리더는 특정 인식 거리에 내에 존재하는 대량의 태그를 동시에 인식한다. 이러한 RFID 기술은 현재 공장 자동화, 재고 관리, 교통, 물류, 유통, 의료 등 다양한 분야에서 널리 활용이 되고 있다[3-5]. RFID 시스템은 태그의 정보를 얻기 위하여 다른 RFID 시스템을 접근하여 처리하는 상황이 발생하게 되는데 이때 각각의 시스템들의 호환성이 문제가 되어 적절한 표준이 필요하게 되었다. 현재 EPCglobal에서는 ALE

· 이 논문은 2006년도 정부(과학기술부)의 재원으로 한국과학재단의 지원(No. R01-2006-000-1080900)과 2007년도 정부재원(교육인적자원부 학술연구조성사업비)으로 한국학술진흥재단의 지원(KRF-2007-314-D00221)을 받아 연구되었음

† 학생회원 : 충북대학교 정보통신공학과  
fredick@netdb.cbnu.ac.kr  
\*\* 학생회원 : 한국과학기술원 전산학과 연구원  
ksbok@dbserver.kaist.ac.kr  
\*\*\* 정 회 원 : 충북대학교 정보통신공학과 교수  
yjs@chungbuk.ac.kr

논문접수 : 2007년 7월 19일  
심사완료 : 2008년 1월 17일

Copyright©2008 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지 : 데이터베이스 제35권 제2호(2008.4)

(Application Level Event)기반의 EPCglobal Architecture Framework을 표준으로 제안하였다[6].

RFID 시스템에서 리더에 의해 인식된 태그 정보를 기본 이벤트라 하며 리더는 다량의 태그를 연속적으로 인식하기 때문에 기본 이벤트는 스트림 데이터 특성을 갖는다. 응용 서비스에서 필요로 하는 이벤트는 기본 이벤트가 아닌 기본 이벤트의 조합으로 이루어진 비즈니스 이벤트이다. RFID 미들웨어는 대용량의 기본 이벤트를 검사하고 응용 서비스에서 요구하는 비즈니스 이벤트를 검출하는 과정을 수행한다[7-9]. 대용량의 기본 이벤트로부터 응용 서비스에서 요구하는 비즈니스 이벤트를 검출하기 위해 미들웨어는 특별한 처리 기법이 필요하다. 비즈니스 이벤트를 검출하기 위해 응용 서비스는 비즈니스 이벤트를 미들웨어에 등록한다. 미들웨어는 연속적으로 발생하는 기본 이벤트를 검사하여 응용 서비스에서 등록한 비즈니스 이벤트가 발생할 경우 이를 응용 서비스에 전달하는 과정을 수행한다. 이러한 과정은 연속 질의(Continuous Query) 혹은 XML 데이터 정제(XML Data Filtering) 처리와 유사한 방식으로 이루어진다[10-12].

E. Wu는 응용 서비스에서 등록한 비즈니스 이벤트를 검출하기 위하여 비즈니스 이벤트의 패턴을 오토마타 형태로 구성하는 SASE를 제안하였다[10,11]. SASE는 등록된 비즈니스 이벤트마다 기본 이벤트를 저장하는 스택이 필요하기 때문에 많은 저장 공간을 낭비한다. 또한 SASE는 순차적인 연산만을 지원하기 때문에 실제 응용 서비스에서 요구하는 다양한 비즈니스 이벤트를 처리하지 못하는 문제점이 있다. F. Wang은 비즈니스 이벤트의 패턴을 그래프 형태로 구성하고 발생하는 모든 기본 이벤트를 그래프의 상위 노드로 전달하여 조건을 비교하는 연산을 수행하는 RCEDA을 제안하였다[12]. RCEDA는 발생하는 모든 기본 이벤트를 상위노드로 전달하여 비교 연산을 수행하기 때문에 기본 이벤트의 수가 증가함에 따라 비즈니스 이벤트를 검출하기 위한 연산 비용이 크게 증가하며 대용량의 기본 이벤트가 발생할 경우 비즈니스 이벤트를 실시간적으로 검출하지 못하는 문제점이 있다.

본 논문에서는 대용량의 RFID 데이터 스트림에 대한 효율적인 비즈니스 이벤트 처리를 위한 새로운 기법을 제안한다. 제안하는 기법은 비즈니스 이벤트를 질의 형태로 등록하고 기본 이벤트가 발생하면 질의 색인을 통하여 비즈니스 이벤트에 접근한다. 질의는 비즈니스 이벤트를 구성하는 기본 이벤트의 발생 여부만을 비교하고 비즈니스 이벤트를 구성하는 기본 이벤트가 모두 발생할 때 실제 비즈니스 이벤트 검출 연산을 수행한다. 이를 통해 비즈니스 이벤트 검출 연산의 수를 감소시킨

다. 또한 과거의 기본 이벤트 발생 이력 정보를 효과적으로 관리하여 비즈니스 이벤트 검출 연산 시 비교해야 할 기본 이벤트의 수를 감소시킨다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문에서 사용하는 용어에 대해 정의하고 RFID 비즈니스 이벤트를 처리하기 위한 기존 연구에 대해 기술한다. 3장에서는 기존의 연구에 대한 문제점을 해결하기 위해 본 논문에서 제안하는 비즈니스 이벤트 처리 기법에 대해 설명한다. 4장에서는 제안하는 이벤트 처리 기법과 기존에 연구된 기법과의 성능 비교를 통해 제안하는 기법의 우수성을 입증한다. 마지막으로 5장에서는 본 논문의 결론을 기술한다.

## 2. 관련연구

### 2.1 용어 정리

RFID 이벤트는 실제 리더에서 인식되는 태그 정보를 나타내는 기본 이벤트와 응용 서비스에서 필요로 하는 의미론적인 이벤트를 나타내는 비즈니스 이벤트로 구분된다.

#### 정의 1. 기본 이벤트(Primitive Event)

기본 이벤트는 특정 시간에 특정 리더에서 인식된 태그 정보를 나타내는 이벤트이다. 기본 이벤트는 중복적으로 발생할 수 있으며 중복된 기본 이벤트는 미들웨어에서 정제된다.

기본 이벤트는 리더의 식별자, 인식된 태그의 식별자, 태그가 리더에 의해 인식된 시간으로 구성된다. 예를 들어, 특정 태그가 부착된 상품 O가 건물 입구에 설치된 리더 R에서 t 시간에 인식되었다면 (R, O, t)라는 3차원의 값으로 나타낸다[12]. 이러한 기본 이벤트는 응용 서비스에서 요구하는 비즈니스 이벤트를 구성하는 최소 단위로 기본 이벤트의 조합을 통해 비즈니스 이벤트를 정의한다.

#### 정의 2. 비즈니스 이벤트(Business Event)

비즈니스 이벤트는 기본 이벤트의 조합으로 구성된 의미적인 이벤트이다. 비즈니스 이벤트는 다수의 연산자를 통해 기본 이벤트를 조합하여 응용 서비스에서 요구되는 이벤트를 정의한다.

비즈니스 이벤트를 조합하기 위해 사용되는 기본 이벤트는 현재 시점에서 발생한 기본 이벤트와 과거 시점에 발생한 기본 이벤트 모두를 포함할 수 있다. 따라서 비즈니스 이벤트를 구성하기 위해 과거에 발생한 기본 이벤트의 정보를 유지해야 한다. 또한 과거의 기본 이벤트들 간의 다양한 조합이 발생할 수 있기 때문에 복잡한 처리 과정이 요구된다.

#### 정의 3. 이벤트 연산자(Event Operator)

이벤트 연산자는 기본 이벤트 조합을 통해 비즈니스

이벤트를 생성하기 위해 필요한 연산자이다.

이러한 이벤트 연산자는 이벤트들이 함께 발생하는 AND( $\wedge$ ) 연산과 하나 혹은 그이상의 발생을 의미하는 OR( $\vee$ ) 연산, 특정 이벤트가 발생하지 않는 NOT(!) 연산, 구성된 이벤트들이 순차적으로 발생하는 SEQUENCE(:) 연산, 마지막으로 특정 시간 내에서 순서에 상관없이 발생하는 INTERVAL(:) 연산자가 있다.

## 2.2 기존 연구

비즈니스 이벤트는 기본 이벤트들 간에 시간 관계, 기본 이벤트의 발생 유무 및 상호 관계에 따라 의미를 가지게 된다. 기존의 SQL은 이벤트간의 시간 관계를 검사하는데 적합하지 않았다. E. Wu는 순차적으로 발생하는 이벤트에서 의미 있는 비즈니스 이벤트를 검출하기 위한 방법으로 SASE라는 언어를 제안하였다[10,11]. SASE는 비즈니스 이벤트를 정의하는 언어로써 SQL과 유사한 구조를 가진다. SASE는 비즈니스 이벤트를 구성하는 이벤트 패턴(EVENT PATTERN), 조건(WHERE) 그리고 시간간격(WITHIN)으로 구성된다. 이벤트 패턴은 기본 이벤트들의 조합으로써 AND, OR, NOT, SEQ의 연산자를 통해 조합된다.

SASE에서 비즈니스 이벤트 검출을 위해 SSC(Sequence Scan and Construction) 연산, Selection 연산, WITHIN 연산 그리고 NG(Negation) 연산을 수행한다. SSC 연산은 비즈니스 이벤트를 오토마타의 형태로 구성하고 발생하는 이벤트를 순차적인 스캔을 통하여 스택에 저장한다. 또한 기본 이벤트의 순차적인 발생에 따라 오토마타의 상태를 변화시킨다. 이때, 오토마타의 상태가 최종 단계에 도달하면 스택을 역으로 탐색하며 후보 집합을 구성한다.

SASE는 등록된 여러 비즈니스 이벤트들 사이에 정보를 공유 할 수 없는 단점을 가지고 있다. SASE에서 각각의 비즈니스 이벤트는 스택에 기본 이벤트를 저장하고 발생한 시간에 따라 포인터를 연결하는 구조를 가지고 있다. 따라서 동일한 기본 이벤트로 구성되어 있는 비즈니스 이벤트도 각각의 기본 이벤트를 저장하고 있어야 한다. 이는 데이터의 중복 저장을 의미한다. 또한 SASE는 비즈니스 이벤트의 패턴이 SEQUENCE 연산으로 한정되어 수행된다. SASE는 비즈니스 이벤트를 오토마타의 형태로 정의한다. 그리고 기본 이벤트가 발생함에 따라 비즈니스 이벤트의 상태를 단계별로 나누어 변화시킨다. 이벤트의 상태는 발생하는 기본 이벤트의 순서에 따라 결정되므로 순서를 고려하는 SEQUENCE 연산만이 오토마타로 구성될 수 있다. SEQUENCE 이외의 연산은 오토마타의 형태로 구성과 비즈니스 이벤트의 상태를 단계별로 정의하는 명확한 기준을 제공하지 못한다. 따라서 SASE는 다양한 형태의 비

즈니스 이벤트를 처리할 수 없는 문제점이 있다.

F. Wang은 비즈니스 이벤트를 효율적으로 처리하기 위하여 비즈니스 이벤트를 정의의 규칙과 검출 방안으로 RCEDA(RFID Complex Event Detection Algorithm)를 제시하였다[12]. RCEDA에서 응용 서비스에 의해 정의된 비즈니스 이벤트는 기본 이벤트와 연산자 그래프들의 조합으로 구성된다. 그래프의 단말 노드에는 기본 이벤트가 위치하고 비단말 노드에는 이벤트를 연결하는 연산자가 위치한다. 그래프의 최상위 노드로 이벤트가 전달되면 완벽한 비즈니스 이벤트가 발생함을 의미한다.

RCEDA에서는 비즈니스 이벤트를 그래프의 형태로 등록하고 발생하는 모든 기본 이벤트는 그래프의 상위 노드로 전달되어 비즈니스 이벤트 검출을 수행한다. 이는 발생하는 기본 이벤트의 수가 증가함에 따라 연산량이 크게 증가되는 단점이 있다. 비즈니스 이벤트는 기본 이벤트의 조합으로 이루어지기 때문에 비즈니스 이벤트가 발생하기 위해서는 비즈니스 이벤트를 구성하는 기본 이벤트가 모두 발생해야 한다. 본 논문에서는 비즈니스 이벤트를 구성하는 모든 기본 이벤트가 모두 발생한 시점을 비즈니스 이벤트의 최소 조건을 만족한다고 한다. RCEDA는 비즈니스 이벤트의 최소 조건을 만족하지 못하는 상황에서도 비즈니스 이벤트를 추출하기 위한 연산을 수행한다. 이때, 비즈니스 이벤트 검출 연산은 완벽한 비즈니스 이벤트를 생성하지 못하고 중간 단계의 후보 집합만을 연산할 수 있다. 중간 단계의 후보 집합은 비즈니스 이벤트를 구성하는 다른 기본 이벤트의 발생 유무에 의해 최종 비즈니스 이벤트의 후보 집합으로 생성하거나 삭제한다. 삭제되는 중간 단계의 후보 집합은 불필요한 연산을 의미하기 때문에 불필요한 중간 단계의 후보 집합을 만드는 과정은 시스템의 자원을 낭비한다.

RCEDA에서 기본 이벤트가 발생하면 하위 노드에서 상위 노드로 전달된다. 이벤트를 전달 받은 상위노드는 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 비즈니스 이벤트의 발생에 영향을 주지 않는 불필요한 중간 단계 후보 집합도 상위 노드로 전달되어 저장하기 때문에 많은 저장 비용을 소모한다. 불필요한 중간 단계의 후보 집합은 상위 노드에서 또다시 불필요한 중간 단계의 후보 집합을 생성한다. 이는 비즈니스 이벤트 추출 연산에서 다시 비교 연산을 수행하는 대상이 되기 때문에 비즈니스 이벤트 검출 시간을 증가시키는 원인이 된다. RCEDA는 여러 비즈니스 이벤트들의 그래프 공유에 대

한 문제점이 발생한다. 비즈니스 이벤트는 서로 다른 비즈니스 이벤트와 그래프의 노드를 공유할 수 있다. 그러나 연산을 구성하는 연산자와 하위노드 연산 시간이 모두 일치 하지 않으면 공유할 수 없다. 따라서 그래프의 단말 노드들은 공유가 용이하지만 비 단말노드는 쉽게 공유되지 않는다는 문제점이 있다.

### 3. 제안하는 비즈니스 이벤트 검출 기법

#### 3.1 제안하는 비즈니스 이벤트 검출 구조

본 논문에서는 기존 비즈니스 이벤트 검출 기법의 문제점을 해결하기 위하여 비즈니스 이벤트의 후보 집합을 추출하는 연산을 수행하기 전에 비즈니스 이벤트의 최소 조건을 검사하는 새로운 기법을 제안한다. 그림 1은 제안하는 비즈니스 이벤트의 처리 구조를 나타낸 것이다. 본 논문에서는 비즈니스 이벤트를 비즈니스 질의와 질의 색인에 등록한다. 기본 이벤트가 발생하면 질의 색인을 통하여 비트맵에 이력 정보를 저장하고 비즈니스 질의에게 기본 이벤트의 발생을 전달한다. 비즈니스 질의는 전달받은 기본 이벤트를 통해 최소 조건을 검사하고 조건을 만족하면 후보 집합을 검출하는 연산을 수행한다. 이때, 후보 집합을 검출하는 과정에서 비트맵을 통하여 과거에 발생한 기본 이벤트의 이력 정보를 참조한다.

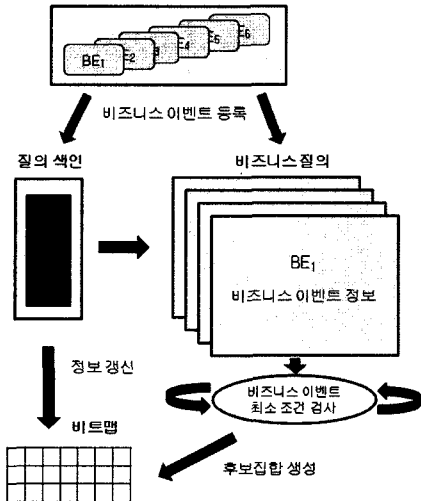


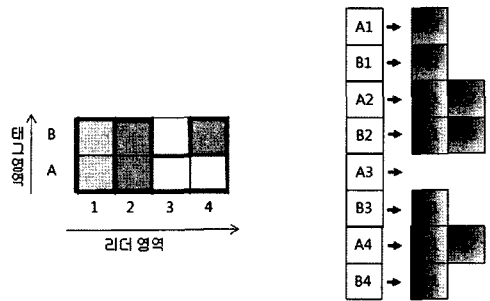
그림 1 제안하는 기법의 전체 구조

##### 3.1.1 질의 색인

질의 색인은 기본 이벤트의 발생을 비즈니스 이벤트에게 빠르게 전달하기 위하여 사용한다. 기본 이벤트를 전달하기 위하여 기본 이벤트의 영역을 논리적인 그리드로 나누고 해당 그리드 영역에 비즈니스 이벤트의 식

별자를 등록한다. 본 논문에서는 태그 영역의 전체 도메인과 리더 영역의 전체 도메인을 논리적인 그리드 영역으로 사용한다. 그리드 영역은 태그와 리더의 그룹을 기준으로 각각의 셀로 나누게 된다. 각각의 셀에는 해당 셀을 참조하는 비즈니스 질의의 식별자가 등록되어 비즈니스 질의를 색인하는 구조로 사용되며 등록된 비즈니스 질의들 사이에서 사용되는 기본 이벤트를 효과적으로 공유할 수 있게 한다.

그림 2는 질의 색인 구조를 나타낸다. 그림 2(a)는 전체 태그와 리더의 도메인 영역을 논리적인 그리드 영역으로 분할된 것을 나타낸다. 예를 들어, 리더 영역의 1은 출구에 설치된 리더 그룹을 나타내며 2는 대여 및 반납실에 설치된 리더 그룹, 3은 도서를 비치하는 열람실에 설치된 리더 그룹, 4는 출구에 설치된 리더 그룹을 의미한다. 태그 영역의 A는 도서를 나타내는 태그 그룹이며 B는 인증된 사용자를 나타내는 태그 그룹이다. 그림 2(b)는 각 그리드 영역에 등록된 비즈니스 질의의 식별자를 저장하는 구조를 나타낸다. Q1은 도서 대여를 의미하는 비즈니스 질의이고 Q2는 도서 반납을 의미하는 비즈니스 이벤트이다. 그림 2(b)의 구조를 통하여 기본 이벤트가 발생하면 해당 비즈니스 질의의 식별자를 통하여 비즈니스 질의에게 기본 이벤트를 전달한다. 그림 2에서와 같이 하나의 셀은 다른 비즈니스 이벤트들 사이에서 공유될 수 있다.



(a) 논리적인 그리드 영역 (b) 비즈니스 질의 등록  
그림 2 질의 색인 구조

##### 3.1.2 비즈니스 질의

비즈니스 질의는 응용 서비스에서 등록한 비즈니스 이벤트를 효율적으로 처리하기 위하여 비즈니스 이벤트의 정보를 저장하는 구조이다. 비즈니스 질의는 비즈니스 이벤트를 기본 이벤트를 사용하여 트리 구조로 등록하고 각각의 기본 이벤트의 최종 발생시간을 저장한다. 기본 이벤트의 최종 발생 시간은 비즈니스 이벤트의 최소 조건 검사에 사용되며 비즈니스 이벤트의 트리 구조는 후보 집합의 연산에 사용된다.

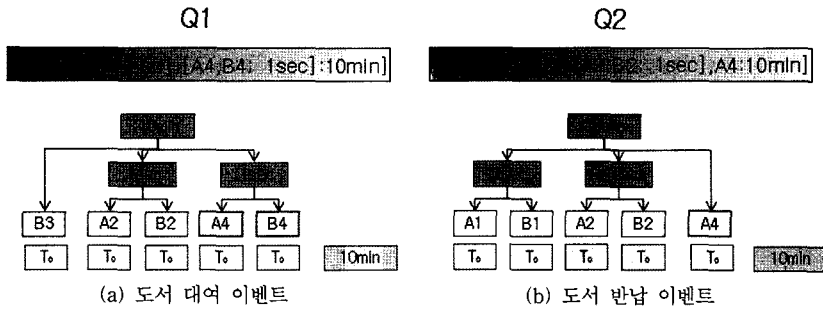


그림 3 비즈니스 이벤트

비즈니스 질의의 구조는 그림 3과 같다. 그림 3(a)는 도서 대여를 의미하는 비즈니스 이벤트, 그림 3(b)는 도서 반납을 의미하는 비즈니스 이벤트가 비즈니스 질의로 등록된 것을 나타낸다. 비즈니스 이벤트의 트리 구조는 RCEDA의 그래프와 유사한 구조를 가지며 중위 순회를 통해 메모리에 유지된다. 각각의 기본 이벤트는 그리드 영역에서의 해당 위치 식별자와 비즈니스 이벤트의 최소 조건을 검사하기 위한 시간 정보를 저장한다. 비즈니스 이벤트를 구성하는데 있어 해당 셀의 기능을 정의하는 다양한 셀 유형이 존재한다. 셀 유형은 일반 셀(Normal Cell), 트리거 셀(Trigger Cell), 부정 셀(Negative Cell), 가상 셀(Image Cell) 등 4가지로 구분된다. 일반 셀은 가장 기본이 되는 셀로 비즈니스 이벤트를 구성하는 보통 셀을 의미한다. 트리거 셀은 비즈니스 이벤트의 조건을 검사하는 시발점이 되는 셀이다. 가상 셀은 비즈니스 이벤트를 분리하여 중간 단계를 하나의 가상 비즈니스 이벤트로 간주하여 나타내는 셀이다. 부정 셀은 발생하지 않는 이벤트, 즉 NOT 연산으로 구성된 셀을 의미한다. 이에 대한 자세한 설명은 3.1.3절에서 기술한다.

3.1.3 셀 유형

비즈니스 질의에서 기본 이벤트의 유형은 역할에 따

라 일반 셀(Normal Cell), 부정 셀(Negative Cell), 트리거 셀(Trigger Cell)로 나눌 수 있다. 또한, 비즈니스 이벤트의 효율적인 처리를 위하여 등록된 비즈니스 이벤트의 단계를 나누어 저장하는 가상 셀(Virtual Cell)이 있다. 일반 셀은 비즈니스 이벤트를 구성하면서 트리거 셀과 부정 셀, 가상 셀에 포함되지 않는 모든 셀을 의미한다. 트리거 셀은 비즈니스 이벤트를 완료할 수 있는 셀을 의미한다. 트리거 셀은 최상의 노드에서 단말 노드까지 확장되며 연산자에 따라 정의된다. SEQUENCE 연산의 경우 연산자에 하위 셀들 중 마지막에 셀이 트리거 셀이 되고 SEQUENCE 연산 이외의 연산자에 의하여 연결된 하위 셀들은 모두 트리거 셀이 된다. 부정 셀은 비즈니스 이벤트의 최소 조건을 검사하는데 사용되지 않는 셀을 의미한다. 부정 셀 역시 최상위 노드에서 단말 노드까지 확장되며 NOT 연산자에 의해 결정된다.

그림 4는 트리에서 사용되는 트리거 셀과 부정 셀의 예를 나타낸 것이다. 그림 4(a)는 트리거 셀을 나타낸다. 비즈니스 이벤트 트리의 최상위 노드는 OR(V) 연산으로 구성된다. OR(V) 연산의 자식 노드는 모두 트리거 셀이 된다. 따라서 OR(V) 연산의 오른쪽 자식인 A3 셀은 단말 노드이기 때문에 트리거 셀이 된다. 하지만 왼

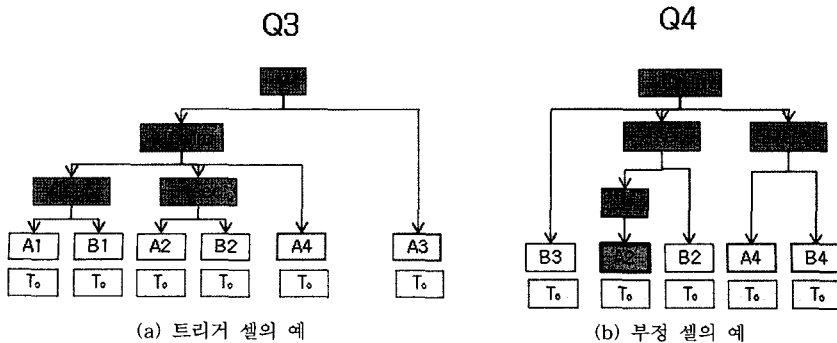


그림 4 트리거 셀, 부정 셀의 예

쪽 자식인 :10min연산은 마지막 자식만을 트리거 셀로 등록하는 연산자이다. 그러므로 :10min의 마지막 자식 노드인 A4는 트리거 셀이 되며 나머지 자식은 일반 셀로 등록된다. 그림 4(b)는 부정 셀의 예를 나타낸다. 그림 4(b)는 도서의 도난을 의미하는 비즈니스 이벤트이다. 그림 3(a)의 도서대여 이벤트와 유사하나 A2의 셀이 NOT 연산자로 구성되어 있다. 이는 도서가 대여 및 반납실에서 인식되지 않고 출구를 벗어남을 나타낸다. 이와 같이 인식되지 않은 이벤트를 나타내기 위하여 부정 셀을 사용한다.

OR 연산으로 연결된 셀은 하나의 셀만 발생하여도 조건을 만족한다. 본 논문에서는 비즈니스 이벤트의 최소 조건을 검사하기 위하여 기본 이벤트의 발생 여부를 검사한다. 즉, 하나 이상의 자식 노드가 조건을 만족하면 상위 노드는 조건을 만족함으로 OR 연산은 비즈니스 이벤트를 나누어야 한다. 가상 셀은 OR 연산으로 연결된 셀들을 가상의 비즈니스 이벤트로 등록을 한다. OR 연산으로 구성된 상위 셀은 가상 비즈니스 이벤트를 의미하는 가상 셀로 구성된다. 또한, 가상 셀은 비즈니스 이벤트를 효율적으로 처리하기 위해 사용한다. 가상 셀은 제안하는 비즈니스 이벤트의 검출 기법에서 발생하는 문제점을 해결하기 위한 방법으로 등록된 비즈니스 이벤트를 두 개 혹은 그 이상의 이벤트로 나누어 처리하기 위한 방법이다. 가상 셀에 관한 자세한 설명은 3.2.3절에서 기술한다.

3.1.4 비트맵

본 논문에서 비트맵을 이용하여 기본 이벤트의 발생 이력을 저장한다. 비트맵은 논리적인 그리드 영역의 셀 하나를 비트열로 나타내어 논리적인 그리드 영역에 대응하는 비트 열을 생성한다. 이러한 비트열은 등록된 비즈니스 이벤트의 연산 시간 중 가장 긴 시간동안 유지되며 (그리드 영역×최대 연산 시간)의 2차원 배열의 구조를 가진다. 비즈니스 질의에서는 기본 이벤트의 최종 발생 시간만을 저장하였다. 과거에 발생한 기본 이벤트의 이력 정보는 비즈니스 질의에서 유지되지 않는다. 따라서 비즈니스 이벤트를 검출하기 위하여 과거에 발생한 기본 이벤트의 이력을 조사해야 한다. 기본 이벤트가 발생하면 그리드 영역을 통과하는 기본 이벤트는 발생한 시간대에 일치하는 비트열에 해당 그리드 영역에 대응하는 비트를 갱신한다. 이러한 비트열은 비즈니스 이벤트의 후보 집합을 생성할 때 사용된다.

비트 열은 단순히 어느 시간대에 어떤 셀의 이벤트가 발생했는지의 정보만을 유지하고 실제 발생한 기본 이벤트는 디스크에 저장되게 된다. 즉, 비트맵은 디스크에 기록된 비즈니스 이벤트를 검색하기 위한 인덱스로 사용된다. 그림 5는 그림 2(a)의 논리적인 그리드 영역에

	A1	B1	A2	B2	A3	B3	A4	B4
T <sub>0</sub>	0	1	0	0	0	0	0	0
T <sub>1</sub>	1	0	0	0	0	1	0	0
T <sub>2</sub>	1	0	0	0	0	0	0	0
T <sub>3</sub>	0	1	1	0	1	0	0	0
T <sub>4</sub>	1	1	0	0	0	0	0	0
T <sub>5</sub>	0	0	0	0	0	0	0	1
T <sub>6</sub>	0	0	1	1	0	0	1	0
T <sub>t</sub>	1	0	0	0	0	0	0	0

그림 5 기본 이벤트 발생 이력에 대한 비트맵

대응하는 비트맵의 구조를 나타낸다. 논리적인 그리드 영역에 이벤트가 발생하면 '1' 발생하지 않으면 '0'의 값을 가지게 된다. T0 시간에서는 B1의 이벤트가 발생하고 T1 시간에서는 A1과 B3 이벤트, Tt 시간에서는 A1의 이벤트가 발생했음을 의미한다.

3.2 제안하는 비즈니스 이벤트 검출 과정

3.2.1 비즈니스 이벤트의 최소 조건을 검사

기존 RCEDA는 기본 이벤트가 발생할 때마다 비즈니스 이벤트를 검출하기 위한 연산을 수행하는 문제점을 가지고 있다. 제안하는 기법에서는 비즈니스 이벤트의 최소 조건을 검사한 후 조건을 만족하는 경우에만 비즈니스 이벤트를 검출 연산을 수행한다. 비즈니스 이벤트의 최소의 조건을 검사하기 위하여 기본 이벤트가 발생하면 질의 색인을 통하여 기본 이벤트가 등록된 비즈니스 질의에서 대응하는 셀의 최종 발생 시간을 갱신한다.

만약 이벤트가 발생된 셀이 해당 비즈니스 이벤트의 트리거 셀이 아니면 비즈니스 이벤트의 최소 조건을 검사하지 않고 트리거 셀일 경우에는 비즈니스 이벤트의 조건을 만족 여부를 검사한다. 비즈니스 이벤트의 최소 조건은 비즈니스 질의를 구성하는 기본 셀 중 부정 셀을 제외한 모든 셀이 비즈니스 이벤트의 최종 시간범위 안에서 발생하였는지 검사한다. 비즈니스 이벤트의 최소 조건을 만족한다고 해서 비즈니스 이벤트가 생성되지 않는다. 단지 비즈니스 이벤트가 생성될 수 있는 기본 조건을 만족했을 뿐이며 비트맵을 검사하여 실제 비즈니스 이벤트의 후보 집합을 찾는 연산을 수행한다.

그림 6은 비즈니스 이벤트의 조건을 검사하기 위한 처리 과정을 나타낸다. B4의 기본 이벤트는 질의 색인을 통해 자신을 포함하는 비즈니스 질의인 Q1에 접근한다. Q1에 접근한 기본 이벤트는 해당 셀에 기본 이벤트가 발생한 시간을 기록한다. 이때 B4의 기본 이벤트는 Q1의 트리거 셀이므로 Q1의 비즈니스 이벤트 최소 조건을 검사한다. 그림 6에서 B3과 A2의 기본 이벤트는 비즈니스 이벤트의 최종 연산 시간인 10분 내에 포함된

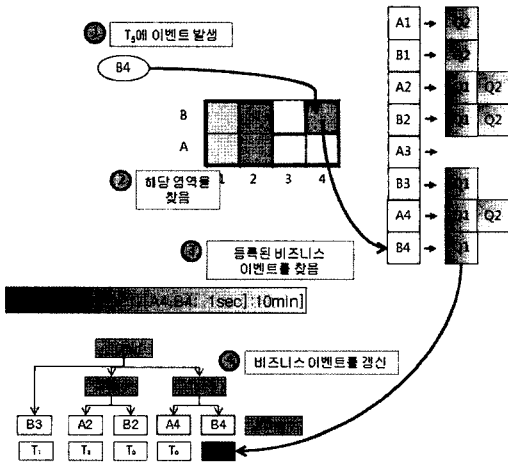


그림 6 비즈니스 이벤트 검사를 위한 처리 과정

```

1 algorithm event_Detection(event_list) //이벤트 처리과정
2 BIT nowBit = 공백 비트열 생성
3 while(event_list 검사)
4   발생한 이벤트를 nowBit에 할당
5 end while
5 while(nowBit의 길이에 대해)
6   if(bit가 1일 때)
7     해당 bit에 대응하는 그리드를 찾고 그리드 영역에
      등록된 비즈니스 이벤트의 값을 갱신
8     if(비트가 비즈니스 이벤트의 트리거 셀일 경우)
9       check_query(Qid)
      //비즈니스 이벤트의 조건을 검사
10      if(검사 조건을 만족할 경우)
11        make_event_set(Qid)
        //비즈니스 이벤트를 생성
12      end if
13    end if
14  end while
15  발생한 모든 이벤트를 디스크에 저장
    
```

그림 7 비즈니스 이벤트 검출 알고리즘

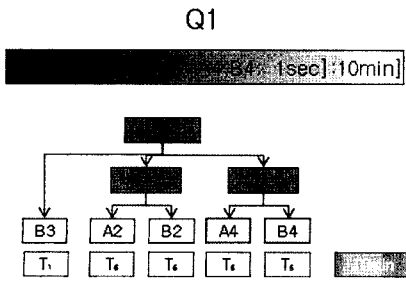
다. 하지만 B2와 A4는 초기 값이 설정된 이후 값이 갱신되지 않았다. 이는 B2와 A4의 기본 이벤트가 발생하

지 않았음을 의미하며 비즈니스 이벤트의 검사 조건을 만족하지 않기 때문에 비즈니스 이벤트의 검출 연산은 수행되지 않는다. 그림 7은 비즈니스 이벤트를 검출하는 알고리즘을 나타낸 것이다.

3.2.2 비즈니스 이벤트의 후보 집합 생성

비즈니스 이벤트의 최소 조건의 만족은 실제 비즈니스 이벤트의 발생을 의미하지 않는다. 단지 비즈니스 이벤트를 구성하는 기본 이벤트들이 모두 발생하여 비즈니스 이벤트를 검출할 최소한의 기본 조건을 만족했음을 의미한다. 비즈니스 이벤트를 검출하기 위하여 제안하는 기법에서는 비트맵을 이용한다. 조건을 만족한 비즈니스 이벤트는 조건이 만족된 시간으로부터 해당 비즈니스 질의의 최종 연산시간 동안의 비트맵을 검사한다. 이때, 검사의 순서는 비즈니스 질의로 구성된 트리를 이용한다. 트리를 구성하는 각각의 기본 이벤트는 상위 노드의 연산 시간의 범위로 슬라이딩 윈도우를 적용하여 해당 연산자의 후보 집합을 탐색한다. 슬라이딩 윈도우를 통하여 탐색된 후보 집합들은 상위 노드로 전달하며 이러한 과정을 반복적으로 수행한다.

그림 8은 슬라이딩 윈도우를 적용하는 연산을 나타낸 것이다. 그림 8(a)는 비즈니스 이벤트를 검출하기 위한 최소 조건이 만족된 경우를 나타낸다. T6 시간에 Q1은 비즈니스 이벤트의 최소 조건을 만족하고 비즈니스 이벤트를 검출하는 연산을 수행한다. 이때, Q1의 트리 구조에 따라 해당 셀에 연산자들의 시간 값을 가지는 슬라이딩 윈도우를 이용하여 후보 집합의 검출한다. Q1에서 비즈니스 질의의 후보 집합을 찾기 위하여 처음으로 A4와 B4의 조합을 비트맵에서 검사하게 된다. ① 연산자의 조건에 만족하는 후보 집합이 있으므로 다음으로는 A2와 B2의 조합에서 후보 집합을 검사한다. ② 이 조합 역시 만족하는 후보 집합이 있으므로 상위의 연산자에 해당하는 후보 집합을 연산한다. 그림 8(b)에서 ①, ② 연산을 통해 생성된 후보 집합은 ③ 연산에서 순차적인 조건을 만족해야 한다. 그림 8에서 A2와 B2의 조



(a) 비즈니스 이벤트

	A1	B1	A2	B2	A3	B3	A4	B4
T <sub>0</sub>	0	1	0	0	0	0	0	0
T <sub>1</sub>	1	0	0	0	0	1	0	0
T <sub>2</sub>	1	0	0	0	0	0	0	0
T <sub>3</sub>	0	1	1	0	1	0	0	0
T <sub>4</sub>	1	1	0	0	0	0	0	0
T <sub>5</sub>	0	0	0	0	0	0	0	1
T <sub>6</sub>	0	0	1	1	0	0	1	0

(b) 비트맵

그림 8 비트맵에 적용된 슬라이딩 윈도우의 예

합이 A4와 B4의 조합보다 나중에 발생하므로 최종 비즈니스 이벤트로 검출되지 않는다.

3.2.3 제안하는 비즈니스 이벤트의 확장

RCEDA에서는 비즈니스 이벤트의 발생을 검출하기 위하여 비즈니스 이벤트의 최소 조건을 만족하지 못하는 경우에도 이벤트 추출 연산을 수행하였다. 이는 불필요한 연산을 증가시키지만 실제 비즈니스 이벤트가 발생하는 상황에서는 연산을 미리 수행하여 결과를 저장하는 장점이 있다. 제안하는 기법에서는 비즈니스 이벤트의 검출을 최소 조건을 만족하는 시점까지 연산을 지연시키고 중간의 후보 집합을 생성하지 않기 때문에 이벤트의 검출 과정을 처음부터 재수행하는 문제가 발생한다. 이러한 문제를 해결하기 위하여 본 논문에서는 가상 셀을 이용하여 가상의 비즈니스 이벤트를 등록한다.

그림 9는 제안하는 기법의 문제점과 해결 방안을 나타낸다. 그림 9(a)는 가상 셀을 사용하지 않는 경우에 발생하는 문제점을 나타낸다. 제안하는 기법에서는 그림 9(a)와 같이 최상위 노드를 기준으로 트리가 불균형적으로 구성된 경우 다음과 같은 문제가 발생한다. A3 이벤트 3개에 대하여 비즈니스 이벤트를 추출하기 위하여 최상위 노드의 왼쪽 자식의 트리의 연산을 반복적으로 수행하는 단점이 있다. 이는 최종 비즈니스 이벤트를 검출하기 위한 연산의 시간을 증가시키는 원인이 된다. 제안하는 기법에서는 가상 셀의 개념을 이용하여 이를 효율적인 처리한다. 그림 9(b)는 Q3의 비즈니스 질의의 :10min 노드를 L\_Q3'의 가상 셀을 이용하여 분할한 구조를 나타낸다. 그림 9(b)의 :10min 노드는 하나의 완전한 비즈니스 이벤트 L\_Q3'로 등록이 된다. 또한 전체 Q3의 비즈니스 이벤트는 L\_Q3'의 가상 셀과 A3의 셀로 구성된다.

가상 셀이 생성되면 가상 셀에서 발생한 정보를 저장하기 위한 가상 비트맵이 생성된다. 가상 셀은 하나의 비즈니스 질의와 동일하게 연산되며 가상 비즈니스 질의에서 결과가 발생하면 가상 비트맵에 발생한 이벤트를 기록한다. 가상 셀은 다시 Q3의 비즈니스 이벤트를 구성하는 셀이 되고 가상 셀은 역시 일반 셀, 트리거 셀, 부정 셀의 유형을 가지게 된다. 가상 셀을 사용하여 구성된 비즈니스 이벤트 역시 다른 비즈니스 이벤트들과 동일한 처리 과정을 수행하게 된다.

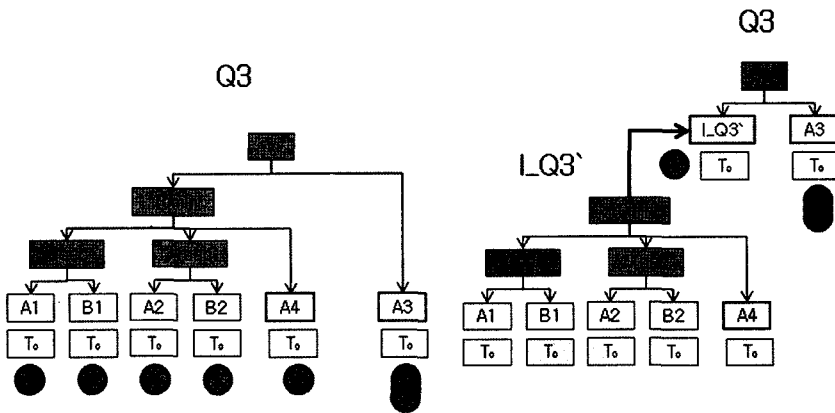
4. 성능 평가

4.1 실험 환경

SASE 기법은 Sequence 연산 이외에 다양한 비즈니스 이벤트의 연산을 포함하는 비즈니스 이벤트를 처리하지 못하는 단점이 있다. 따라서 제안하는 비즈니스 이벤트 처리기법의 성능을 평가하기 위하여 RCEDA 기법과의 비교 평가를 수행한다. 성능 평가에 사용된 시스템은 Pentium(R) 4 2.8GHz 프로세서와 1Gbyte의 메모리를 가지고 있으며 운영체제는 Microsoft Windows XP Professional을 사용한다. 비즈니스 이벤트를 처리를 위한 모듈은 C++를 이용하여 Microsoft Visual Studio .Net 2003에서 개발하였다. 성능 평가를 수행하기 위해 다양한 요소들을 변화시키면서 수행하였으며 성능 평가 요소는 표 1과 같다. 성능 평가에서는 각각의 요소 값들을 변경하면서 비즈니스 이벤트의 평균 연산 수행 시간, 연산 횟수, 지연 시간 등을 측정한다.

4.2 비즈니스 이벤트 복잡도(C)에 따른 성능 평가

이 절에서는 비즈니스 이벤트의 복잡도에 따른 성능 평가를 수행하기 위해 비즈니스 질의를 구성하는 트리의 레벨 C를 변경하면서 비즈니스 이벤트 처리 시간과



(a) 가상 셀 미적용 시 문제점

(b) 가상 셀을 적용한 예

그림 9 가상 셀을 사용하는 예

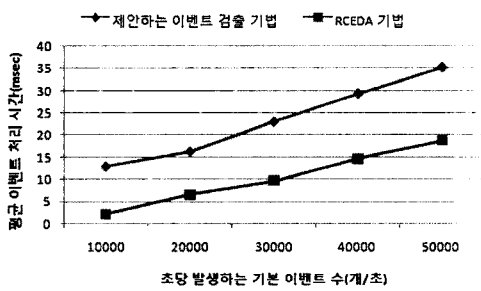


표 1 성능평가 요소

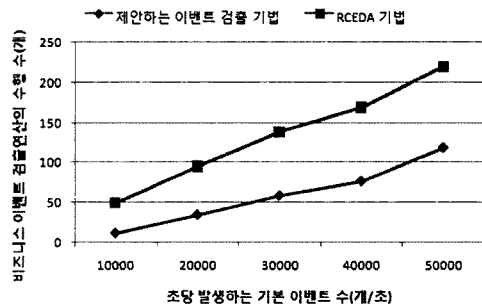
성능 평가 요소	설명
C	비즈니스 이벤트의 복잡도 (트리 레벨)
N	등록된 비즈니스 이벤트 수
P	초당 발생하는 기본 이벤트 (이벤트 수/초)

연산 수를 평가한다. 실험을 위한 비즈니스 이벤트는 트리 형태로 구성되며 C는 트리의 레벨을 의미한다. 이 절에서는 트리를 2진 균형 트리로 구성하고 N을 500개로 등록하여 성능 평가를 수행한다.

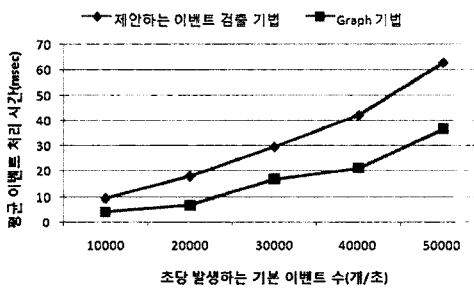
그림 10은 비즈니스 이벤트의 복잡도에 따른 성능 평가 결과를 나타낸 것이다. 그림 10(a)와 그림 10(b)는 C가 1인 경우 비즈니스 이벤트를 처리하는데 소요되는 시간과 연산 수를 나타낸 것이다. 그림 10(a)에서 제안하는 기법과 RCEDA 기법 모두 P가 증가함에 따라 선형적으로 연산 시간이 증가함을 볼 수 있다. 트리의 레벨이 1인 경우 제안하는 기법이 RCEDA 기법과 비교하여 10msec의 연산 시간이 더 소요됨을 볼 수 있다. 이는 제안하는 기법이 RCEDA 보다 비즈니스 이벤트의 최소 조건을 검사하기 위한 연산을 추가적으로 수행하



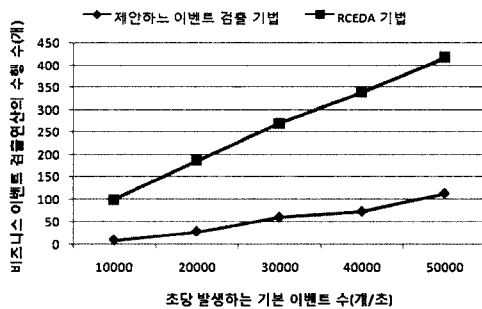
(a) 트리 레벨 1에서 평균 처리 시간



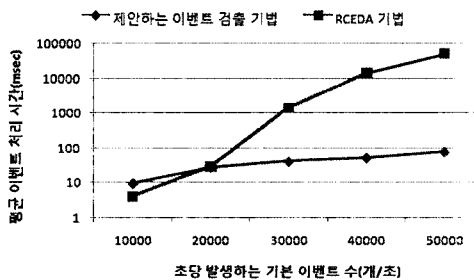
(b) 트리 레벨 1에서 연산 수



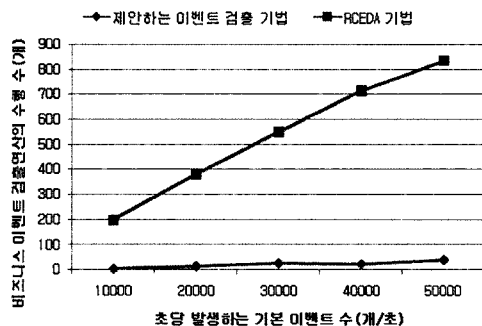
(c) 트리 레벨 2에서 평균 처리 시간



(d) 트리 레벨 2에서 연산 수



(e) 트리 레벨 3에서 평균 처리 시간



(f) 트리 레벨 3에서 연산 수

그림 10 비즈니스 이벤트 복잡도에 따른 성능 평가

기 때문이다. 또한 비즈니스 이벤트의 검출과정에서 중간 단계의 후보 집합을 항상 새로 연산하기 때문에 더 많은 시간이 소요되기 때문이다. 비즈니스 이벤트의 복잡도가 작을수록 비즈니스 이벤트의 최소 조건은 쉽게 만족되고 비즈니스 이벤트의 검출 연산도 간단하기 때문이다. 그림 10(b)는 P의 변화에 따라 RCEDA에서 처리하는 비즈니스 이벤트 수와 제안하는 기법에서 비즈니스 이벤트의 최소 조건을 만족하여 검출 과정을 처리하는 비즈니스 이벤트 수이다. 이 실험에서 비즈니스 이벤트의 최소 조건을 약 50%의 이벤트가 만족함을 볼 수 있었다.

C가 2인 경우는 그림 10(c), 그림 10(d)와 같다. 그림 10(c)에서 보여주는 것과 같이 제안하는 기법은 기존 RCEDA에 비해 더 많은 연산 시간이 소요한다. 이는 비즈니스 이벤트의 최소 조건을 처리하는 과정이 모든 비즈니스 이벤트를 처리하는 과정보다 연산 시간이 더 소모됨을 나타낸다. 하지만 P가 증가함에 따라 제안하는 기법보다 RCEDA가 연산 시간의 증가율이 더 큰 것을 볼 수 있다. 그림 10(d)는 P의 변화에 따라 RCEDA에서 처리하는 비즈니스 이벤트 수와 제안하는 기법에서 비즈니스 이벤트의 최소 조건을 만족하여 검출 과정을 처리하는 비즈니스 이벤트 수를 나타낸다. P가 증가함에 따라 RCEDA는 처리하는 비즈니스 이벤트의 수가 크게 증가하는 반면 제안하는 기법은 비즈니스 이벤트의 최소 조건을 연산하여 약 20~25% 정도로 비즈니스 이벤트 검출 연산의 수를 감소시킨다.

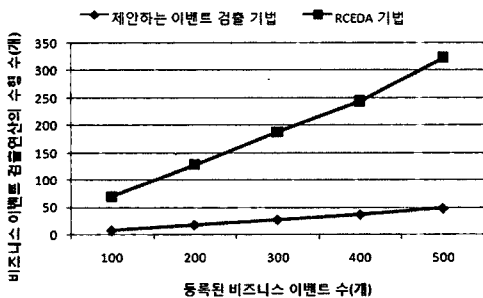
C를 3으로 설정하여 위와 같은 실험을 계속한 결과는 그림 10(e), 그림 10(f)와 같다. 그림 10(e)에서 보는 것과 같이 RCEDA은 P가 증가함에 따라 평균 처리 시간이 지수적으로 증가하지만 제안하는 기법은 선형적으로 증가한다. 또한, 비즈니스 이벤트를 처리하는 평균 시간이 P가 30,000개인 시점을 기준으로 1초를 벗어나는 것을 볼 수 있다. 이는 비즈니스 이벤트의 복잡도가 높아지면 기본 이벤트의 발생에 따라 접근되는 비즈니스 이

벤트의 수가 크게 증가하기 때문이다. 불필요한 중간 단계의 후보 집합이 증가하면서 실제 비즈니스 이벤트를 추출하는 연산에서 시간 지연을 발생시키기도 한다. 그림 10(f)는 P의 변화에 따라 RCEDA에서 처리하는 비즈니스 이벤트 수와 제안하는 기법에서 비즈니스 이벤트의 최소 조건을 만족하여 검출 과정을 처리하는 비즈니스 이벤트 수를 나타낸다. P의 증가에 따라 RCEDA는 비즈니스 이벤트 검출 연산을 수행하는 비즈니스 이벤트의 수가 크게 증가한다. 하지만 제안하는 기법은 검출 연산을 수행하는 비즈니스 이벤트의 수가 크게 증가하지 않는다. 이는 등록된 비즈니스 이벤트가 복잡도가 증가할수록 제안하는 기법이 기존의 RCEDA 기법보다 빠른 이벤트 처리를 수행한다는 것을 나타낸다.

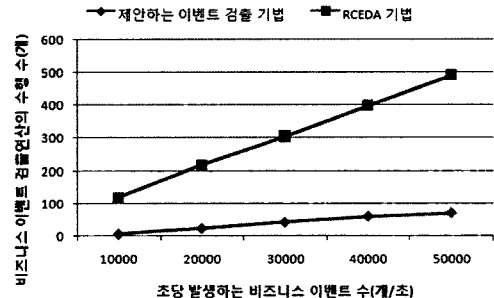
4.3 처리되는 이벤트 수

이 절에서는 초당 30,000개의 기본 이벤트가 발생하는 상황(P=30,00)에서 비즈니스 이벤트의 수(N)를 100~500개로 변경시켜 등록하고 RCEDA에서 처리하는 비즈니스 이벤트 수와 제안하는 기법에서 검출 연산을 수행하는 비즈니스 이벤트의 수를 측정한다. 이때, 비즈니스 이벤트의 복잡도는 1~3까지의 임의의 값으로 설정한다. 또한 N을 500개로 고정한 후 P를 10,000~50,000개로 변경시키며 성능 평가를 반복한다.

그림 11(a)은 N의 변화에 따라 실험 평가를 수행한 결과이다. 초당 30,000개의 기본 이벤트가 발생할 때 제안하는 기법은 RCEDA에 비하여 10~15% 정도 감소한 수의 비즈니스 이벤트 검출 연산을 수행한다. 즉, 이벤트의 복잡도가 다양하면 본 논문에서 제안하는 기법이 처리하는 비즈니스 이벤트 검출 연산의 수가 현저히 감소함을 볼 수 있다. 그림 11(b)는 P를 변화시키며 제안하는 기법과 RCEDA 기법을 비교한 결과이다. P가 증가함에 따라 RCEDA와 제안하는 기법 모두 초당 발생하는 이벤트에 따라 처리해야 할 연산 수가 증가됨을 볼 수 있다. 하지만 P의 증가에 따라 제안하는 기법이 RCEDA 기법보다 증가량이 적음을 알 수 있다. RCEDA



(a) 비즈니스 이벤트 수에 따른 연산 수



(b) 초당 발생하는 기본 이벤트 수에 따른 연산 수

그림 11 비즈니스 이벤트 처리 수에 따른 성능 평가

기법은 증가된 기본 이벤트에 따라 대응하는 비즈니스 이벤트의 수가 증가한다. 하지만 제안하는 기법은 기본 이벤트의 수가 증가하여도 비즈니스 이벤트의 최소 조건을 만족하는 비즈니스 이벤트의 수는 크게 증가하지 않는다.

4.4 평균 수행시간

이 절에서는 N과 P를 변화에 따라 초당 이벤트를 처리하는 평균 시간을 측정한다. 실험은 4.3절과 동일한 환경에서 수행한다. 그림 12(a)는 등록된 N에 따른 평균 처리 시간을 나타낸 것이다. RCEDA 기법은 N에 따라 처리 시간이 비교적 크게 증가하는 것을 볼 수 있다. 반면 제안하는 기법은 N에 따라 평균 처리 시간이 크게 증가하지 않는다. 그림 11(a)에서와 같이 N이 증가함에 따라 RCEDA 기법에서 처리해야 하는 비즈니스 이벤트의 수는 크게 증가하나 제안하는 기법은 증가율이 적기 때문에 처리 시간 역시 유사한 형태로 나타난다.

그림 12(b)는 P의 변화에 따른 비즈니스 이벤트의 평균 처리 시간을 나타낸다. 이번 실험 역시 그림 11(b)에서와 같이 제안하는 기법을 통해 조건을 만족하는 비즈니스 이벤트의 수가 현저히 감소하기 때문에 P를 증가시켜도 제안하는 기법의 처리시간은 변화가 크게 증가되지 않는다. 하지만 RCEDA 기법에서는 P가 증가함에 따라 비즈니스 이벤트를 처리하는 시간이 급격히 증가함을 볼 수 있다. 또한 초당 40,000개의 기본 이벤트가 발생하게 되면 처리 시간이 1초를 넘게 되어 이벤트 처리에 지연 시간이 발생함을 볼 수 있다.

4.5 지연시간

4.4절에서는 이벤트가 처리되는 평균 시간에 대하여 성능 평가를 수행하였다. 하지만 평균 시간은 이벤트 처리가 지연되는 시간을 나타내기에는 부족하다. 1초당 이벤트가 발생하는 상황에서 연산이 1초 이전에 끝났더라도 아직 들어오지 않은 이벤트를 처리할 수 없기 때문이다. 따라서 이 절에서는 각 시간대별 이벤트 처리 시간을 계산한다. 연산 시간이 1초 이하인 경우는 지연

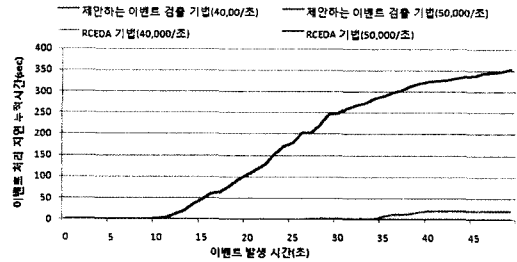


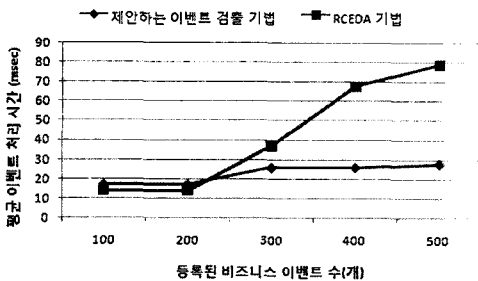
그림 13 이벤트 발생 시간대별 이벤트 처리 지연 시간

시간이 발생하지 않는 것이나 1초 이상인 시간은 모두 지연시간에 포함된다. 임의의 비즈니스 이벤트 복잡도 C를 갖는 비즈니스 이벤트 500개를 등록하고 성능평가를 수행한다. 각각의 비즈니스 이벤트들은 레벨에 따라 총 연산 시간이 다르나 최대 30초를 넘지 않으며 비즈니스 이벤트의 최대 연산시간을 포함할 수 있는 50초 동안 실험을 수행하였다.

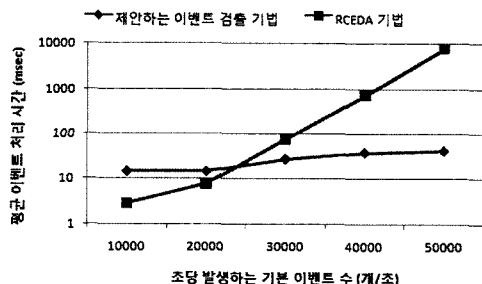
그림 13은 각 이벤트 처리 시간대별 이벤트 처리의 지연 시간을 누적한 것을 나타낸 것이다. 그림 13에서 보는 것과 같이 이벤트의 발생 초기에는 지연 시간이 발생하지 않지만 특정 시점에서 초기 지연이 발생하기 시작하면서 일정 기간 동안 지연 시간이 증가한다. 지연 시간의 증가는 점차 완만히 발생하는 것을 볼 수 있다. RCEDA 기법에서 기본 이벤트가 40,000개 발생할 때 약 20초 정도의 지연이 발생하며 기본 이벤트가 50,000개 발생할 때 약 35초 정도의 지연이 발생한다. 그러나 제안하는 기법은 지연 시간이 거의 발생하지 않는다.

5. 결론 및 향후 연구

본 논문에서는 비즈니스 이벤트 검출을 효율적으로 처리하기 위해 비즈니스 이벤트의 최소 조건을 검사하고 조건을 만족하는 시점에서 비즈니스 이벤트를 검출하는 방법을 제안하였다. 제안하는 비즈니스 이벤트 검출 기법은 비즈니스 이벤트를 구성하는 기본 이벤트의



(a) 비즈니스 이벤트 수에 따른 평균 처리 시간



(b) 초당 발생하는 기본 이벤트 수에 따른 평균 처리 시간

그림 12 비즈니스 이벤트의 평균 처리시간

발생 조건을 빠르게 판별하기 위해 질의 색인을 구성하고 비즈니스 이벤트를 구성하는 기본 이벤트를 트리 구조로 등록하여 관리한다. 비즈니스 이벤트를 구성하는 모든 기본 이벤트가 발생하면 비트맵을 이용하여 비즈니스 이벤트는 후보 집합을 생성하게 된다. 제안하는 비즈니스 이벤트는 비즈니스 이벤트를 만족하는 최소 조건을 만족할 때 비즈니스 이벤트 검출 연산을 수행하기 때문에 연산의 수를 감소시킬 수 있고 불필요한 후보 집합을 생성하지 않는다. 성능 평가를 통해 제안하는 기법이 기존의 RCEDA 기법에 비해 복잡도가 높은 비즈니스 이벤트 처리에 우수한 성능을 나타냈으며, 초당 많은 양의 기본 이벤트가 발생하는 상황에서도 우수한 성능을 나타냈다. 향후 연구 방향으로 본 논문의 내용을 실제 응용 서비스에서 발생하는 데이터에 적용하여 실제 응용에서의 효율성을 입증할 예정이다. 또한 분산 환경에서의 비즈니스 이벤트의 검출 연산 처리에 대한 연구를 수행할 예정이다.

### 참 고 문 헌

- [1] 박석지, 유종현, "u-센서 네트워크 산업의 개념과 발전 동향", IITA 주간기술동향, 제1135호, pp. 1-19, 2004.
- [2] 유승화, "RFID 기술 현황 및 활용분야", 정보과학회지, 제23권, 제7호, pp. 64-70, 2005.
- [3] F. Wang and P. Liu, "Temporal Management of RFID Data," Proc. International Conference on Very Large Data Bases, pp. 1128-1139, 2005.
- [4] 박승창, "RFID 서비스 기술의 최근 국내의 연구개발 동향과 전망", 한국통신학회지(정보통신), 제23권, 제12호, pp. 70-80, 2006.
- [5] Y. Hu, S. Sundara, T. Chorma, and J. Srinivasan, "Supporting RFID-based Item Tracking Applications in Oracle DBMS Using a Bitmap Datatype," Proc. International Conference on Very Large Data Bases, pp. 1140-1151, 2005.
- [6] EPCglobal, The EPCglobal Architecture Framework Final Version, 2005.
- [7] 정태수, 김영일, 이용준, "RFID 미들웨어 플랫폼 기술", TELECOMMUNICATIONS REVIEW, 제15권, 제2호, pp. 290-307, 2005.
- [8] M. Mohania, D. Swamini, S. K. Gupta, S. Bhowmick, and T. Dillon, "Event Composition and Detection in Data Stream Management Systems," Proc. International Conference on Database and Expert Systems Applications, pp. 756-765, 2005.
- [9] M. K. Moon, Y. B. Kim, and K. H. Yeom, "Contextual Events Framework in RFID System," Proc. International Conference on Information Technology : New Generations, pp. 586-587, 2006.
- [10] E. Wu, Y. Diao, and S. Rizvi, "High Performance Complex Event Processing over Streams," Proc. ACM SIGMOD International Conference on Management of Data, pp. 407-418, 2006.
- [11] D. Gyllstrom, E. Wu, H. J. Chae, Y. Diao, P. Stahlberg, and G. Anderson, "SASE: Complex Event Processing over Streams(Demo)," Proc. Biennial Conference on Innovative Data Systems Research, pp. 407-411, 2007.
- [12] F. Wang, S. Liu, P. Liu, and Y. Bai, "Bridging Physical and Virtual Worlds: Complex Event Processing for RFID Data Streams," Proc. International Conference on Extending Database Technology, pp. 588-607, 2006.



노진석

2005년 8월 충북대학교 정보통신공학과 졸업 공학사. 2007년 8월 충북대학교 정보통신공학과 졸업 공학석사. 2007년 8월~현재 대우증권 비즈니스시스템부 사원. 관심분야는 데이터베이스 시스템, 위치기반 서비스, RFID 등



복경수

1998년 2월 충북대학교 수학과 이학사  
2000년 2월 충북대학교 정보통신공학과 공학석사. 2005년 2월 충북대학교 정보통신공학과 공학박사. 2005년 3월~2008년 2월 한국과학기술원 정보전자연구소 연수연구원. 2008년 3월~현재 가인정보 기술 기술제안팀 차장. 관심분야는 자료 저장 시스템, 멀티미디어 데이터베이스, 이동객체 데이터베이스, 센서네트워크 및 RFID 등

유재수

정보과학회논문지 : 데이터베이스  
제 35 권 제 1 호 참조