

# WS-ECA 수리 모형

이강찬\* · 이원석\* · 이승윤\* · 신동민\*\*

\*한국전자통신연구원, \*\*한양대학교

## The Mathematical Model for WS-ECA

Kangchan Lee\* · Wonsuk Lee\* · Jonghong Jeon\* · Seungyun Lee\* · Dongmin Shin\*\*

\*Electronics and Telecommunications Research Institute, \*\*Hanyang University

E-mail : {chan, wslee, syl}@etri.re.kr\* · dmshin@hanyang.ac.kr

### ABSTRACT

본 논문은 유비쿼터스 환경에서 다수의 다양한 이벤트가 모여 구성되는 ECA 룰 조건을 이용하여 동작하는 디바이스를 분석하였다. 그리고 디바이스에서 처리하는 시간제약이 있는 ECA 룰 조건 처리모형을 분석하고 디바이스가 동작을 처리하는 평균처리시간을 구하였다. 또한 디바이스가 셋 이상의 ECA 룰을 처리하는 경우에 대해 모형화하고 평균 ECA 룰 조건 처리시간을 구하였다.

### 키워드

Web Services, ECA rule, ubiquitous 디바이스, CFA

### I. 서론

유비쿼터스 환경에서 시스템을 구성하는 디바이스들의 실제 동작은 각 디바이스들이 동작을 수행하기 위한 이벤트를 획득했을 때 이루어지며 이러한 이벤트는 하나 이상의 데이터 소스(data source)에서 획득이 가능하다. 또한 유비쿼터스 환경에서는 모든 동작에 대한 처리가 실시간으로 이루어지며 이러한 동작의 근원이 되는 이벤트는 확정적으로 발생하지 않고 랜덤한 양상을 띠게 된다. 이러한 이벤트가 모여 유비쿼터스 환경의 디바이스들을 동작 가능하게 하는 이벤트들의 집합을 조건(condition)라고 정의한다.

유비쿼터스 환경은 기존의 다양한 네트워크를 모두 포함하므로 다양한 디바이스들에 의해 구성된다. 또한 유비쿼터스를 구성하는 각 디바이스들의 동작은 시스템 내의 각 디바이스들이 생성 또는 전달하는 이벤트에 의해 구성되는 ECA 룰의 조건에 의해 유기적으로 동작한다. 이러한 ECA 룰의 조건은 단순하고 단일한 이벤트로 구성될 수도 있고 다수의 복잡한 이벤트로 구성된 형태일 수도 있다.

본 논문에서는 유비쿼터스 환경에서 다수의 다양한 이벤트가 모여 구성되는 ECA 룰 조건을 이용하여 동작하는 디바이스를 분석하였으며, 디바이스에서 처리하는 시간제약이 있는 ECA 룰 조건

처리모형을 분석하고 디바이스가 동작을 처리하는 평균처리시간을 구하였다. 또한 디바이스가 셋 이상의 ECA 룰을 처리하는 경우에 대해 모형화하고 평균 ECA 룰 조건 처리시간을 구하였다.

### II. 유비쿼터스 환경에서 ECA 룰의 시간제약이 있는 조건

유비쿼터스 환경에서 발생하는 대부분의 이벤트는 영구데이터가 아니다. 어떤 이벤트가 한 디바이스에게 의미가 있기 위해서는 제한된 시간 내에 처리되어야 하며 해당 이벤트가 가지는 시간 제약 안에 디바이스에게 제공되지 못하거나, ECA 룰의 조건을 만족하지 못하는 경우 해당 이벤트는 그 의미를 상실하게 되고 버려지게 된다. 그러므로 시스템이 복잡해지고 시스템을 구성하는 디바이스의 수가 많아질수록 이러한 시간제약을 만족시킬 수 있도록 시스템을 구성하는 것이 필요하다.

### III. ECA 룰의 시간제약이 있는 조건 처리

한 디바이스가 구성한 ECA 룰의 조건에 의해 동작한 결과는 다른 디바이스의 이벤트로 사용할 수도 있다. 그러므로 디바이스는 다른 디바이스의 ECA 룰의 조건을 구성하는 이벤트를 제공하는 데이터 소스로 시스템 내에 존재할 수도 있고 다른 디바이스로부터 이벤트를 받아 ECA 룰의 조

건을 구성하여 독립적인 동작을 수행할 수도 있다. 또한 단순히 이벤트를 다른 디바이스에게 전달하는 역할만을 수행할 수도 있다. 이러한 디바이스들의 동작들은 모두 시스템을 구성하는 디바이스들을 통해 생성, 전달되는 이벤트로 구성되는 ECA 룰 조건에 의해 이루어진다. 시스템내의 각 디바이스가 동작한다는 것은 해당 디바이스가 동작하기 위한 ECA 룰의 조건을 구성하였다는 것을 의미한다.

앞서 언급한 ECA 룰의 시간제약이 있는 조건에 의한 디바이스의 동작을 모형화하기 위해 먼저 다음과 같은 가정을 둔다.

- 1) 각 디바이스가 획득하는 이벤트는 포아송과정으로 발생한다.
- 2) 각 디바이스가 ECA 룰의 조건을 만족하고 해당 동작을 수행하는 것은 지수분포를 따르는 집단서비스로 제공된다고 가정한다.
- 3) 이벤트가 디바이스에게 의미를 상실한 경우 디바이스에 존재하는 이벤트의 수는 감소하게 된다. 이벤트가 디바이스에게 의미를 상실하는 경우가 발생하는 시간은 지수분포를 따른다.
- 4) 디바이스가 수행하는 동작은 해당 동작에 대한 ECA 룰의 조건이 필요로 하는 이벤트의 수가 만족되어 완전한 조건을 만족한 동작부터 순서대로 FIFO(First In First Out) 정책을 적용하여 수행한다.
- 5) ECA 룰의 조건을 구성하는데 필요한 이벤트의 수를 만족하여 해당 디바이스가 동작하는 중에 도착하게 되는 이벤트는 해당 디바이스에 의미 없는 이벤트이다.
- 6) 시스템은 open environment로서 디바이스로 도착하는 이벤트의 데이터 소스는 무한하다고 가정한다. 이는 시스템을 구성하는 디바이스의 수가 무한하다는 것과 같은 의미이다.
- 7) 디바이스는 FIFO 정책을 적용하여 동작을 수행하고 주기상에 존재하는 ECA 룰의 조건을 타입으로 구분하는 것이 아닌 동작순서로 구분한다. 그러므로 하나의 주기에서 동일한 타입에 대한 ECA 룰의 조건이 두 개 이상 존재할 수 없다.

또한 유비쿼터스 환경에서 발생하는 대부분의 이벤트는 영구데이터가 아니다. 어떤 이벤트가 한 디바이스에게 의미가 있기 위해서는 제한된 시간 내에 처리되어야 하며 해당 이벤트가 가지는 시간제약 안에 디바이스에게 제공되지 못하거나, ECA 룰의 조건을 만족하지 못하는 경우 해당 이벤트는 그 의미를 상실하게 되고 버려지게 된다. 그러므로 시스템이 복잡해지고 시스템을 구성하는 디바이스의 수가 많아질수록 이러한 시간제약을 만족시킬 수 있도록 시스템을 구성하는 것이 필요하다.

디바이스가 동작하는 데에 필요한 ECA 룰의 조건을 구성하기 위해 만족해야 하는 이벤트의 수

는  $n$ 이라고 한다.  $n$ 개의 이벤트를 만족할 때까지 시간이 경과함에 따라 이벤트의 수가 증가하거나 감소하는 확률과정  $\{N(t), t \geq 0\}$ 을 생각해 보자.  $N(t)$ 는 시점  $t$ 에서의 이벤트 개수이며 이 확률과정의 상태공간은  $E = \{0, 1, 2, \dots, n-1, n\}$ 이 된다. 이때  $T_k$ 를  $k$ 개의 이벤트가 있을 때  $n$ 개의 이벤트가 만족될 때까지의 시간이라고 하면  $E(T_k)$ 는  $k$ 개의 이벤트가 있을 때 ECA 룰의 조건이 만족되기 위한  $n$ 개의 이벤트가 만족될 때까지의 평균시간이 된다. 그러므로 0개의 이벤트가 있을 때  $n$ 개의 이벤트가 만족될 때까지의 평균시간인  $E(T_0)$ 는 ECA 룰 조건이 구성되는 평균시간이 된다. 그림 1은 확률과정  $\{N(t), t \geq 0\}$ 의 전이율다이아그램이다.

결국 ECA 룰의 조건이 만족되기 위한 평균시간인  $E(T_0)$ 와 동작시간의 평균인  $1/\mu$ 을 합한 시간은 이벤트가 도착하기 시작하여 동작을 완료할 때까지의 평균시간이 된다. 이러한 동작을 완료할 때까지의 평균시간은 유비쿼터스 환경에서 각각의 디바이스가 동작을 처리하는 평균시간으로 사용할 수 있다.

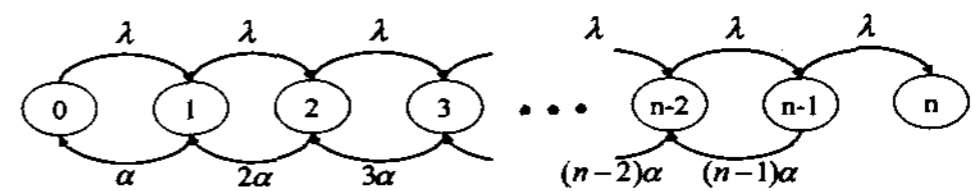


그림 1. 확률과정의 전이율다이아그램

확률과정  $\{N(t), t \geq 0\}$ 의 전이율다이아그램을 통해 조건이 구성되기 위한 평균시간  $E(T_0)$ 를 다음과 같이 구할 수 있다.

$$E(T_0) = \frac{1}{\lambda} \left[ \sum_{j=1}^{n-1} \prod_{k=1}^j \frac{k\alpha}{\lambda} + \sum_{i=2}^{n-1} \sum_{j=i}^{n-1} \prod_{k=i}^j \frac{k\alpha}{\lambda} + n \right]$$

한 시점에 디바이스로 도착한 하나의 이벤트는 두 가지 ECA 룰의 조건에 동시에 사용 가능하다. 또한 각 조건에 의한 수행 가능한 동작도 두 가지 형태를 가진다. 그림 2, 그림 3 그리고 그림 4는 두 가지 형태에 대한 ECA 룰의 조건에 의해 발생 가능한 경우들을 보여준다. 각 그림에서  $n_1$ 은 두 가지 형태의 ECA 룰 중 먼저 조건의 구성요건을 만족한 조건의 이벤트 개수를 의미하고  $n_2$ 는 두 번째로 구성요건을 만족한 조건의 이벤트 개수를 의미한다.

- 1) 첫 번째 조건의 동작이 수행되고 해당 동작을 수행하는 동안 두 번째 ECA 룰의 조건에 대한 이벤트 개수가 만족되는 경우

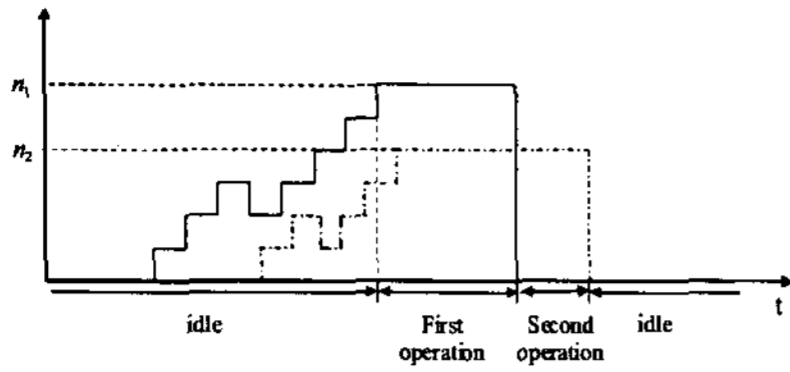


그림 2. 첫 번째 동작 중 다른 조건의 이벤트 개수가 만족된 경우

그림 2는 디바이스가 첫 번째 ECA 룰의 조건을 구성하기 위한 이벤트의 수인  $n_1$ 이 만족되어 첫 번째 동작을 수행하는 첫 번째 동작 상태로 전이되고 이 동안 두 번째 ECA 룰의 조건을 구성하기 위한 이벤트의 수인  $n_2$ 가 만족된 경우를 나타내고 있다. 이 경우  $n_2$ 가 만족된 시점부터 첫 번째 동작이 끝나는 시점까지는 두 번째 동작을 수행하기 위한 기다림의 시간이 되며 첫 번째 동작이 끝나자마자 두 번째 동작을 수행하게 된다.

2) 첫 번째 ECA 룰의 조건이 만족되고 해당 동작을 수행하는 동안 다른 이벤트가 도착하지 않은 경우

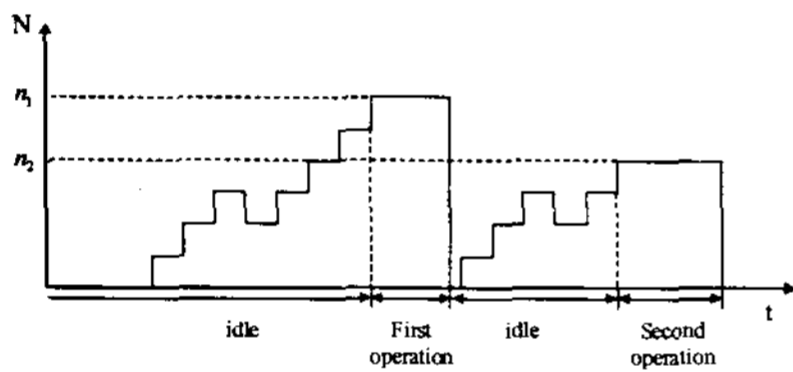


그림 3. 첫 번째 동작 중 다른 ECA 룰의 조건에 대한 이벤트가 도착하지 않은 경우

그림 3은 첫 번째 동작이 끝난 시점에도 아직 두 번째 ECA 룰의 조건에 대한 이벤트가 도착하지 않은 경우를 나타낸다.

3) 첫 번째 ECA 룰의 조건이 만족되고 해당 동작의 수행이 끝난 후에도 두 번째 ECA 룰의 조건에 대한 이벤트개수가 만족되지 못한 경우

그림 4는 첫 번째 ECA 룰에 대한 동작이 끝났지만 아직 두 번째 ECA 룰에 대한 동작을 위한 이벤트의 개수가 만족되지 못한 경우이므로 유힬 상태로 전이하였다가  $n_2$ 가 만족된 시점에 두 번째 동작 상태로 전이하게 된다.

하나이상의 ECA 룰을 처리하는 경우를 고려하기 위해 ECA 룰의 조건에 대한 동작이 끝난 시점에 ECA 룰의 조건을 구성하는 이벤트의 상태에 대해 고려한다. 이전 ECA 룰에 대한 동작이 끝난 시점에서 다음 ECA 룰의 조건 구성을 위한 이벤트의 개수를  $k$ 라 하자. 이때 1)은  $k=n_2$ 인 경우이며(그림 2), 2)는  $k=0$ (그림3), 3)은  $1 \leq k \leq n_2 - 1$ (그림 4)인 경우를 나타낸다. 이전 ECA 룰에 대한 동작이 끝난 시점에 이벤트가  $k$  개 있을 확률을  $P_k$ 라 하자.

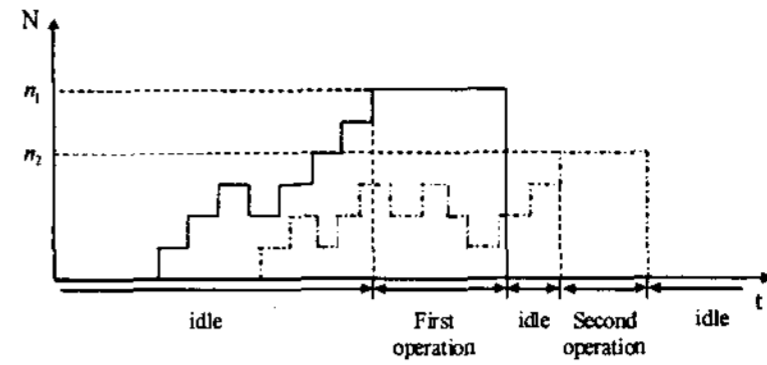


그림 4. 첫 번째 동작 중 다른 ECA 룰의 조건에 대한 이벤트 개수가 만족되지 못한 경우

$n$ 개의 이벤트들을 만족할 때까지 시간이 경과함에 따라 이벤트의 수가 증가하거나 감소하고  $n$ 개의 이벤트가 만족되면 상태 0으로 전이하는 새로운 확률과정  $\{M(t), t \geq 0\}$ 을 생각해 보자.  $M(t)$ 는 시점  $t$ 에서의 이벤트 개수이며 이 확률과정의 상태공간은  $E = \{0, 1, 2, \dots, n-1, n\}$ 이 된다. 그림 27은 확률과정  $\{M(t), t \geq 0\}$ 의 전이율 다이어그램이다. 확률과정  $\{M(t), t \geq 0\}$ 에서 각 디바이스로 도착하는 이벤트의 발생은 발생률  $\lambda$ 의 포아송 과정으로 발생한다. 또한 각 이벤트는 시간이 경과함에 따라 해당 디바이스에게 그 의미를 상실할 수 있으며 그 발생율은  $\alpha$ 이다. 해당 디바이스가 획득한 이벤트의 수가 총  $n$ 개가 되었을 때 ECA 룰의 조건이 만족된다. 구성된 ECA 룰의 조건은 디바이스의 동작을 발생시키고 동작시간은 평균이  $1/\mu$ 인 지수분포를 따른다.

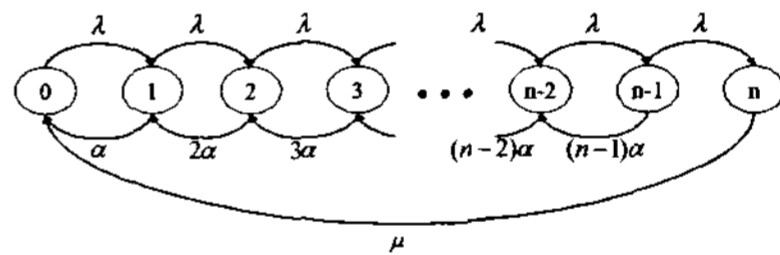


그림 5. 확률과정  $\{M(t), t \geq 0\}$ 의 전이율 다이어그램

그림 5에서 상태  $n$ 으로의 평균 이탈률은  $\mu P_n$ 이고 평균 입력률은  $\lambda P_{n-1}$ 이 된다. 또한 상태  $n-1$ 로의 평균이탈률은  $(\lambda + (n-1)\alpha) P_{n-1}$ 이고 평균 입력률은  $\lambda P_{n-2}$ 이 된다. 확률과정  $\{M(t), t \geq 0\}$ 를 통해 다음과 같이 이전 ECA 룰에 대한 동작이 끝난 시점에 이벤트가  $k$  개 있을 확률을  $P_k$ 를 구할 수 있다.

$$P_k = \frac{\mu}{\lambda} \left[ \sum_{i=0}^{n-2} \left[ \frac{\mu}{\lambda} \sum_{j=i+1}^{n-1} \prod_{l=j+1}^i \frac{j\alpha}{\lambda} \right] + \frac{\mu}{\lambda} (n-1) + 1 \right]^{-1} \left[ 1 + \sum_{i=k+1}^{n-1} \prod_{j=k+1}^i \frac{j\alpha}{\lambda} \right], 0 \leq k \leq n-2,$$

$$P_{n-1} = \frac{\mu}{\lambda} \left[ \sum_{i=0}^{n-2} \left[ \frac{\mu}{\lambda} \sum_{j=i+1}^{n-1} \prod_{l=j+1}^i \frac{j\alpha}{\lambda} \right] + \frac{\mu}{\lambda} (n-1) + 1 \right]^{-1}.$$

한편,  $E(T_k)$  값을 앞서 구한  $E(T_0)$ 의 값을 이용하여 다음과 같이 구할 수 있다.

$$E(T_k) = \frac{1}{\lambda} \left[ \sum_{j=k}^{n-1} \prod_{l=1}^j \frac{l\alpha}{\lambda} + \sum_{i=2}^{n-1} \sum_{j=k}^{i-1} \prod_{l=i}^j \frac{l\alpha}{\lambda} + (n-k) \right], 1 \leq k \leq n-1.$$

그러므로 3)의 경우에 해당하는 첫 번째 동작이 끝난 후 두 번째 ECA 룰의 조건을 위한 이벤트가  $k$  개가 있는 경우 첫 번째 동작이 끝난 후부터 두 번째 동작이 끝날 때까지의 평균시간을

$E(T^*)$ 라 한다면 그 값은 다음과 같다.

$$\begin{aligned}
 E(T^*) &= \sum_{k=1}^{n-1} E(T_k)P_k + E(T_{n-1})P_{n-1} \\
 &= \sum_{k=1}^{n-1} \left[ \sum_{j=k}^{n-1} \prod_{l=j}^{n-1} \frac{\lambda}{\lambda} + (n-k) \right] \\
 &\quad \times \frac{\mu}{\lambda} \left[ \sum_{i=0}^{n-2} \left[ \frac{\mu}{\lambda} \sum_{j=i+1}^{n-1} \prod_{l=j}^{n-1} \frac{\lambda}{\lambda} \right] + \frac{\mu}{\lambda} (n-1) + 1 \right] \left[ 1 + \sum_{i=k+1}^{n-1} \prod_{l=i}^{n-1} \frac{\lambda}{\lambda} \right] \\
 &\quad + \frac{1}{\lambda} \left[ \sum_{j=k}^{n-1} \prod_{l=j}^{n-1} \frac{\lambda}{\lambda} + \sum_{j=2}^{n-1} \prod_{l=j}^{n-1} \frac{\lambda}{\lambda} + (n-k) \right] \times \frac{\mu}{\lambda} \left[ \sum_{i=0}^{n-2} \left[ \frac{\mu}{\lambda} \sum_{j=i+1}^{n-1} \prod_{l=j}^{n-1} \frac{\lambda}{\lambda} \right] + \frac{\mu}{\lambda} (n-1) + 1 \right].
 \end{aligned}$$

이제 앞서 언급한 1) 첫 번째 동작 중 다른 ECA 룰의 조건에 대한 이벤트 개수가 만족된 경우 ( $k=n_2$ ), 2) 첫 번째 동작 중 다른 ECA 룰의 조건에 대한 이벤트 개수가 만족되지 못한 경우 ( $k=0$ ), 3) 첫 번째 동작 중 다른 ECA 룰의 조건에 대한 이벤트가 도착하지 않은 경우 ( $1 \leq k \leq n_2 - 1$ )에 대한 ECA 룰 처리의 평균시간을 종합하여 구할 수 있다.

현재 고려하고 있는 두 가지 형태의 평균 ECA 룰 처리시간  $E(T)$ 는 다음식과 같다.

$$\begin{aligned}
 E(T) &= \sum_{i=1}^2 \left[ E(T^{(i)}) + \frac{1}{\mu_i} \right] = E(T^{(1)}) + E(T^{(2)}) + \frac{1}{\mu_1} + \frac{1}{\mu_2} \\
 &= E(T^{(1)}) + \sum_{k=0}^{n_2} E(T_k^{(2)})P_k + \frac{1}{\mu_1} + \frac{1}{\mu_2}.
 \end{aligned}$$

그림 6은  $m(m \geq 3)$ 가지 형태의 ECA 룰을 처리하기 위한 조건의 이벤트의 개수가 시간의 경과에 따라 변화하는 모습의 한 예를 보여준다. 여기서 ECA 룰이 처리되는 순서를 고려하여 디바이스가 시간제약이 있는 ECA 룰의 조건을 처리하는 평균시간을 구한다.

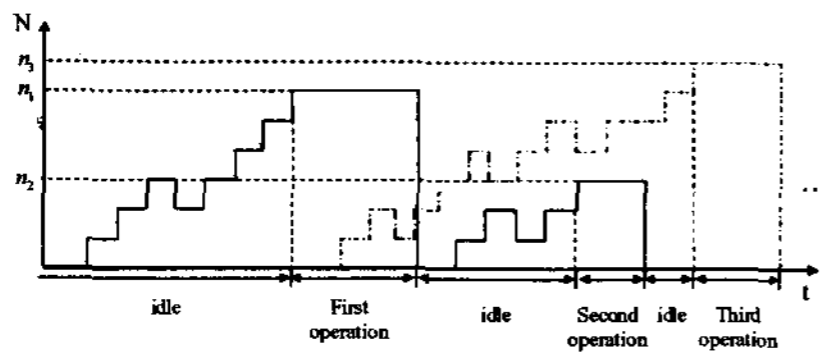


그림 6.  $m(m \geq 3)$ 가지 형태의 ECA 룰의 조건에 대한 이벤트 개수의 변화

결국 한 주기 동안  $m(m \geq 3)$ 가지 형태 ECA 룰을 모두 처리하는데 걸리는 평균시간을  $E(T)$ 라고 하고  $m$ 가지 형태의 ECA 룰이 존재한다고 했을 때  $i$ 번째 처리되는 ECA 룰의 조건에 대한 이벤트가 만족되는 시간과 동작시간을  $E(T^{(i)})$ 와  $1/\mu_i$ 라고 한다면 한 주기 동안 ECA 룰을 처리하는 평균시간  $E(T)$ 는 다음과 같다.

$$\begin{aligned}
 E(T) &= \sum_{i=1}^m \left[ E(T^{(i)}) + \frac{1}{\mu_i} \right] = \sum_{i=1}^m E(T^{(i)}) + \sum_{i=1}^m \frac{1}{\mu_i} \\
 &= E(T^{(1)}) + \sum_{i=2}^m \left[ \sum_{k=0}^{n_i} E(T_k^{(i)})P_k^{(i)} \right] + \sum_{i=1}^m \frac{1}{\mu_i}.
 \end{aligned}$$

#### IV. 결론

유비쿼터스 환경에서 다수의 다양한 이벤트가 모

여 구성되는 ECA 룰 조건을 이용하여 동작하는 디바이스를 분석하였다. 그리고 디바이스에서 처리하는 시간제약이 있는 ECA 룰 조건 처리모형을 분석하고 디바이스가 동작을 처리하는 평균처리시간을 구하였다. 또한 디바이스가 셋 이상의 ECA 룰을 처리하는 경우에 대해 모형화하고 평균 ECA 룰 조건 처리시간을 구하였다.

그리고 이벤트의 개수에 따라 ECA 룰의 질이 결정된다고 할 때 평균처리시간이 시간제약 안에 만족하도록 이벤트의 개수인 ECA 룰의 질과 평균처리시간의 트레이드오프가 필요함을 알 수 있었다.

이러한 결과는 유비쿼터스 환경에서 시간제약이 있는 ECA 룰 조건 처리모형에 적용이 가능하다. 또한 ECA 룰 조건을 구성하는 이벤트들이 가지는 제약이 시간제약뿐만 아니라 잘못된 입력으로 인한 데이터의 손실을 고려하는 모형에서도 적용이 가능하다.

#### References

- [1] Papamarkos, G., A. Poulouvasilis, and P.T. Wood, Event-condition-action rules on RDF metadata in P2P environments. *Computer Networks*, 2006. 50(10): pp. 1513-1532.
- [2] Paton, N.W. and O. Diaz, Active database system. *ACM Computing Survey*, 1999. 31(1): pp. 63-103.
- [3] S. Vinoski. *Integration with Web Services*. *IEEE internet computing*, 7(6): 75-77, 2003.
- [4] A. Carter and M. Vukovic. A Framework For Ubiquitous Web Service Discovery. In *Proc. of the 6th UbiComp*, 2004.
- [5] A. Sashima, N. Izumi, and K. Kurumatani. Location-Mediated Coordination of Web Services in Ubiquitous Computing, in *Proc. of IEEE Int'l Conf. Web Services (ICWS'04)*, pages 109-114, 2004.
- [6] N. Bassiliades, and I. Vlahavas. 디바이스: Compiling production rules into event-driven rules using complex events. *Information and Software Technology*, 39:331-342, 1997.
- [7] K. Liu, L. Sun, A. Dix, and M. Narasipuram. Norm Based Agency for Designing Collaborative Systems. *Information Systems Journal*, 11(3): 229-247, 2001.
- [8] S. Calo, and M. Sloman, Policy-Based Management of Net-works and Services. *Journal of Network and Systems Management*. 11(3):249-252, 2003.