

## 웹기반 분산 기업시스템의 적기협력을 위한 시간 제약 조건의 분석

서태정\* · 이동우\*\*

---

### <국문 초록>

---

최근 기업들은 B2B EC와 같은 웹 기반 분산 환경에서 적기 협력을 필요로 한다. 특히 시간 제약을 필요로 하는 긴급 협조 요청이나 긴급 메시지는 즉시 처리 하여야 한다. 본 논문에서는 웹 기반 분산 시스템 환경에서 기업사이의 적기 협력을 위한 시간 제약 조건을 사건, 데이터, 행위 관점에서 분석하였고, 시스템 설계자들이 시간 제약 조건과 이의 위반에 관련된 적기 협력 방안을 적절히 표현할 수 있는 ECA 규칙 프로그래밍 기법을 제안한다. 그리고 웹 기반 능동 기능 컴포넌트를 이용하여 프로토타입을 구현하였다.

주제어 : 능동 데이터베이스, 기업 사이의 적기협력, ECA 규칙, 시간제약

---

\* 우송정보대학 교수

\*\* 교신 저자: 이동우(dwlee@wsu.ac.kr), 우송대학교 교수, 042-630-9713

## I. 서론

최근의 기업 시스템들은 웹기반 분산 환경에서 운영된다. 이들 기업 시스템들은 기업들의 공동 사업 목적을 달성하기 위하여 서로 긴밀하게 협력하여야 한다. 특히 기업사이의 시간 제약을 필요로 하는 긴급 협조 요청이나 긴급 메시지에 대해서는 즉시처리 협력을 하여야 한다. 그러나 현재 대부분의 시스템들은 이를 적절히 다루지 못하고 있고 주로 ad hoc 방식으로 해결하고 있다.

시스템의 시간 제약성에 관한 연구는 주로 실시간 시스템이나 실시간 데이터베이스 시스템에서 이루어져 왔고, 주로 주어진 시간 제약 조건 하에서 태스크(task)나 트랜잭션의 처리 및 스케줄링에 대한 것이다(Audsley, 1991; Dasarathy, 1985; Purimetla, 1995; Ramamritham, 1993; Son, 1995). 따라서 적기 시간 요구조건은 인위적으로 시스템 설계과정에서 ad hoc 방법으로 시간 제약 조건을 할당하는 방식을 취해 왔다. 그리고 주어진 시간 제약의 의미를 제대로 알지 못하기 때문에 시스템의 시간 제약은 과도하게 주어지기도 한다.

일반적으로 실시간 시스템과 비 실시간 시스템 사이의 구분은 시간이 지남에 따라 무가치(invalid)해지는 데이터, 시기에 맞게 발생해야만 하는 사건(event) 및 시기에 맞게 완료 되어야 하는 행위(action)의 존재 여부에 달렸다. 따라서 시간 제약 조건의 기원 및 그 의미를 파악하여 시스템 설계에 반영하여야 과도한 시간 제약을 지닌 시스템 설계를 피할 수 있다.

Web의 등장과 성장으로 B2B 전자상거래가 크게 활성화되었고 많은 연구가 이루어졌다(Medjahed et. al 2003). 그러나 이들 연구는 주로 B2B의 상호운영성에 대한 연구가 대부분으로, 통신 계층, 내용 계층, 경영 프로세스 계층에 대한 프로토콜을 주로 다루었고, 기업 사이의 적기 협력과 시간 제약에 대한 논의는 없다.

본 논문에서는 웹기반 B2B EC 분산 환경에서 기업사이의 적기협력을 위한 시간 제약조건의 시간적 의미(timing semantics)에 대하여 사건, 데이터, 행위 관점에서 살펴보고, 능동 데이터베이스의 ECA(Event Condition Action) 규칙으로 시간 제약 조건과 관련된 행위로 표현하여 과도한 시간 제약을 지닌 시스템 설계를 피하고 가장 적절한 응용을 설계하는 방안과 웹 분산 환경에서 이를 구현하는 방법을 제안한다.

논문의 나머지 부분은 다음과 같이 구성된다. 다음 장에서는 문제 정의를 위한 B2B EC 환경에서 기업사이의 적기 협력 응용 예를 살펴보고, 3 장에서는 응용에서 시간 제약을 결정하는 요인을 다룬다. 4 장에서는 시간 제약성의 의미를 사건, 데이터, 행위 관점에서 다루고, 5 장에서 이의 ECA 규칙화를 보인다. 6 장에서 능동 기능 컴포넌트를 이용한 프로토타입 구현을 설명하고 7 장에서 결론으로 요약과 앞으로의 연구에 대하여 논한다.

## II. 기업 사이의 적기 협력

문제 정의를 위하여 이 절에서는 B2B EC 분산 환경에서 기업 사이의 적기 협력의 필요성과 ECA 규칙 시스템에 의한 지원을 살펴본다.

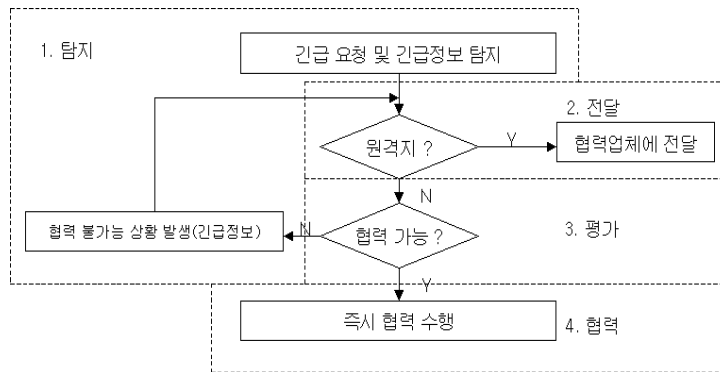
### 1. 적기 협력의 필요성

기업사이의 적기 업무 협력(즉시요청-즉시처리협조)이 필요한 경우는 기업의 통상적인 업무 처리 과정에서 예외적인 경우로 볼 수 있다. 이러한 경우에는 협력 업체에 긴급 협조를 요청하거나 상대방이 필요로 하는 긴급 정보를 즉시 전달(즉시요청)해야 하고 협력 업체는 이에 대하여 즉시 협조(즉시처리협조)해야 하는 경우이다. 이러한 예외적인 상황의 발생은 아주 빈번하지는 않지만, 기업과 고객의 이익에 크게 영향을 미치고 있다.

계약을 통한 계약이행으로 기업들이 상호 협력을 하므로 적기 방식의 협력을 필요로 하는 경우는 다음과 같이 4가지로 분류할 수 있다(이동우, 2005):

- 1) 약속된 협력 업무를 제대로 처리할 수 없는 경우: 협력 업체의 내부 사정으로 업무 협조를 제대로 수행할 수 없는 경우, 이 사실을 협조 서비스를 요청한 업체에 즉시 통보해 주어야 상대방 업체가 대책을 세울 수 있다.
- 2) 약속된 협력 업무의 처리를 보완 또는 수정해야 하는 경우: 약속된 협력 업무 내용 중의 일부를 보완 또는 수정해야 하는 경우.
- 3) 약속된 협력 업무 처리를 취소해야 하는 경우: 기업 내부의 사정에 의하여 요청했던 업무 협조를 취소해야 하는 경우.
- 4) 약속된 일반 협력 업무 처리와 별도로 특수 처리를 해야 하는 경우: 통상적인 업무 처리와 별도로 특별 서비스를 필요로 하는 경우.

위와 같은 적기 협력이 필요한 경우에 대하여 기업들이 서로 어떻게 협력할 것인가를 계약에 새로 추가하여 점진적으로 시스템에 반영할 수 있다.



[그림 1] 기업사이의 적기 협력 절차

## 2. ECA 규칙 시스템에 의한 적기 협력

B2B 전자상거래 환경에서 기업사이의 적기 협력을 위한 절차를 정리하면 [그림 1]과 같다. 이 절차는 다음과 같이 4 단계로 구성 된다(이동우, 2005):

- 1) 탐지 : 한 기업이 다른 기업에 적기 협력을 필요로 하는 긴급 요청을 해야 하거나, 다른 기업에게 즉시 통보해야 할 긴급 정보가 발생한 경우를 인식하는 단계이다.
- 2) 전달 : 통신망을 통하여 탐지된 상황을 협력 업체에 전달하는 단계.
- 3) 평가 : 요청된 협조 업무의 수행가능 여부와 통지된 긴급 정보에 대한 조치 가능성을 평가하는 단계로, 협력 가능 제약 조건을 점검하여야 한다. 협력 업체가 협력을 할 수 없는 경우가 있기 때문이다. 협력 가능 제약 조건에는 제한된 시간 이내에 또는 절차상 시간이 경과하여 협조할 수 없는 시간 제약 조건과 요청한 제품이나 서비스를 실행할 자원이 없어 협조할 수 없는 자원 제약 조건이 있다.
- 4) 협력 : 요청된 협조 업무를 실행하거나 통지된 긴급 정보에 대한 조치를 취하는 단계이다.

B2B 전자상거래 환경에서 기업들이 적기 협력을 하기 위해서는 위와 같은 절차를 따라야 하는데, 이러한 적기 협력 기능을 B2B 전자상거래 시스템에 반영하기 위해서는, 기업 사이에 즉시 처리를 필요로 하는 상황을 탐지하고 이를 서로 통보할 수 있는 기능과 통보된 상황을 인식하고 이에 대한 조치(협력 행위)를 취할 수 있는 기능이 있어야 한다. 이것은 능동데이터베이스의 ECA 규칙 메커니즘에 적합한 응용임을 알 수 있다(Paton, 1999; Widom, 1996). 즉, 협력 업체 사이에서 적기 협력을 필요로 하는 상황을 ECA 규칙에 따라, 상황의 발생을 사건으로, 그리고 이 사건이 즉시 처리를 필요로 하는지 여부를 조건으로, 그리고 이에 대한 조치(협력 행위)를 행

위로 표현할 수 있다. 그러면, ECA 규칙을 실행하는 능동 기능 컴포넌트가 사건의 발생을 자동으로 탐지하여, 이에 대한 조치를 취하는 협력 업체의 시스템에 사건 발생을 즉시 통보하고, 통지 받은 협력 시스템의 능동 기능 컴포넌트는 조건을 검사하여 해당 상황 여부를 판단하여 조치(협력 행위)를 취함으로써 업무 협조를 응용 프로그램이나 사용자의 간섭 없이 즉시 처리 방식으로 구현할 수 있다.

### Ⅲ. 시스템의 시간 제약 조건을 결정하는 요인

앞 절에서 살펴본 응용 예에서 다음과 같은 시간 제약 조건을 결정하는 요인들을 볼 수 있다 :

- 1) 물리적 시스템의 성질. 예, 자동차의 고속도로 주행 속도( 배송시간 계산에 필요), 컨베이어 벨트의 속도( 객체의 분류 시간을 결정) 등.
- 2) 서비스 요구 조건(QoS requirements) : 고객이나 협력 기업이 감내 할 수 있는 서비스 지연.
- 3) 인간의 반응(행위) 시간: 협력 업체 직원이 협조 업무를 인식하고 처리하는 데 걸리는 시간.

위 요인들은 협력시스템이 주변 상황으로부터 지켜야 할 시간제약 조건을 결정하며, 이 요인들 사이에는 상대적인 중요성이 다를 수 있다. 즉, 물리적인 환경 요인 보다 인간에게 관련된 상호 작용 요인이 더 중요할 수 있다. 이러한 정보에 의하여 행위의 우선순위를 결정할 수 있다.

시간 제약 조건 설계상 문제점은 이러한 것들이 대부분 인위적이라는 것이다. 즉, 컨베이어 벨트의 길이나 자동차의 속도는 비용과 기술적인 사항에 의해 결정되고 있으며, 전화망의 서비스 요구의 질은 규정에 의해 정해진다. 그리고 서비스 공급자가 보장하는 조치시간은 비용이나 타사와의 경쟁적인 요소에 의해 결정된다. 이러한 결정은 컴퓨터시스템 설계자의 권한이 아니다. 시스템 설계자는 단지 다른 사람이 정해 놓은 범위 내에서 결정을 할 수밖에 없다. 이것이 외부적으로 주어진 제약이다.

시스템 설계자에게 주어지는 요인들의 수는 가급적 적어야 제약 조건을 만족시킬 수 있는 선택의 여지가 많아진다. 설계자가 결정을 내려야 할 때 또 다른 제약들이 뒤따른다. 사용되는 컴퓨팅 플랫폼(computing platform), 소프트웨어 설계방법, 이미 존재하는 컴포넌트 시스템 등에 의한 것이다. 설계상의 한 단계에서의 결정은 다른 단계에 영향을 미친다.

## IV. 사건, 데이터, 행위에서의 시간적 의미

### 1. 사건의 시간적 의미

시간 제약의 적기 협력 응용에서, 어떤 행위는 어떤 사건의 발생에 의해서 시작된다. 즉, 각 사건의 발생과 각 행위는 연관 관계가 있다. 따라서 행위에 대한 어떤 시간 제약들은 이 행위를 시동시키는 사건들로부터 기인한다. 그러므로 사건들은 이것들과 관련된 시간적 성질과 함께 다루어야 한다. 기본적으로 세 가지 종류가 있다([Dasarathy, 1985].

- 1) 최대(Maximum) : 두 사건 사이의 지연 시간. 예를 들면, 배송차량이 객체들을 꼭 배송해야만 하는 시간 ( $t_1$  시간 이내.)
- 2) 최소(Minimum): 두 사건 사이의 지연 시간. 예로, 연속된 두 객체의 배송 허가는  $t_2$  시간 이내에는 허용되어서는 안 된다.
- 3) 기간(Duration) : 한 사건의 길이. 예를 들면, 기간을 갖는 사건은 복합 사건으로 볼 수 있다. 즉 단순 사건의 시작과 끝으로 볼 수 있다.

사건은 입력 또는 자극(S)사건, 출력 또는 반응(R) 사건, 그리고 내부나 외부 사건으로 구분된다. 최대, 최소, 기간 제약은 자극 사건과 반응 사건들 사이에 발생할 수 있다. (S-R, S-S, R-S, R-R): 일반적인 스케줄에 관한 논문들은 S-R의 경우만 다룬다. 반복해서 발생하는 자극 사건에 관련된 최대와 최소 시간제약을 비율-기반(rate-based) 시간제약이라 한다. 즉 이것은 반복해서 발생하는 사건의 최대 또는 최소 도착 율과 관계있기 때문이다.

한 사건이 발생하면 이것과 관련된 행위가 이를 처리한다. 따라서 사건에 관련된 시간 제약은 행위들의 시간 제약에 영향을 미친다. 즉, 비율-기반 제약은 행위의 주기성을 만든다. 자극 사건과 이것의 반응에 관련된 시간 제약은 마감 시간 제약이 된다. 한 사건의 제약 조건은 이처럼 이 사건을 처리하는 행위의 제약조건이 된다.

### 2. 데이터에서의 시간적 의미

협력 시스템은 주변상황에 관련된 가능한 자료를 근거로 이의 환경과 상호 작용한다. 환경의 상태는 협력 시스템이 인식하고 있는 것과 실제 상태가 일치하여야 한다. 그렇지 않으면 협력 시스템의 결정은 잘못될 수도 있고, 아마도 그 영향은 엄청날 것이다. 따라서 환경에 대한 적기 감시와 수집된 정보의 즉시 처리는 필수적이다.

감지된 데이터의 처리는 새로운 데이터를 만든다. 예를 들면, 배송차량의 위치, 방향, 속도 등은 객체의 도착 시간을 계산하는데 쓰인다. 이 계산은 자동차의 본질과도 관련이 있다. 따라

서 필요한 정보를 저장 기억 장소에서 가져와야 할지도 모른다. 협력 시스템의 반응은 이 계산된 데이터에 달렸다. 즉, 배송 위치추적 제어 시스템의 경우, 이 계산된 정보는 배송차량에게 우회 지시를 보내거나 통과 하도록 한다. 일반적으로 환경의 이력 정보 역시 저장 기억 장소에 기록된다.

협력 시스템이 올바른 결정을 하고 환경에 적절한 조치를 취하기 위해서는 환경에 대한 뷰가 실제 상태와 거의 같아야 한다. 즉, 실제 상태와 데이터베이스가 나타내는 상태가 서로 일관성이 있어야 한다. 이것을 시간적 일관성이라 한다. 시간적 일관성에 두 가지 구성요소가 있다 (Audsley, 1991; Song, 1992).

1) 절대적 일관성(Absolute consistency) : 환경의 상태와 데이터베이스에서의 반영.

데이터베이스의 데이터(data item)를 다음과 같이 정의하자:

$$d : (\text{value}, \text{avi}, \text{timestamp})$$

여기서  $d_{\text{value}}$ 는  $d$ 의 현재 상태,  $d_{\text{timestamp}}$ 는  $d$ 의 관측시간,  $d_{\text{avi}}$ 는  $d$ 의 절대 유효 기간(absolute validity interval)로 timestamp 이후  $d$ 가 절대적으로 유효한 기간을 의미한다.

2) 상대적 일관성(Relative consistency) : 다른 데이터를 유도하는데 사용된 데이터 사이에서 ; 새로운 데이터 항목(data item)을 구하는데 쓰인 데이터 항목들의 집합이 상대적 일관성 집합(relative consistency set)을 형성한다. 그러한 집합  $R$ 에서  $R_{rvi}$  는 상대적 유효 기간(relative valid interval)을 말한다.

$d \in R$ 라고 가정하면,  $d$ 는 다음 두 조건을 만족할 때만 올바른 상태가 된다:

1.  $d_{\text{value}}$  는 논리적으로 일관적이다-데이터에 명시된 모든 무결성 조건을 만족.
2.  $d$ 가 시간적으로 일관적:

$$\text{- 절대적 일관성 : } (\text{Current\_time} - d_{\text{timestamp}}) \leq d_{\text{avi}} .$$

$$\text{- 상대적 일관성 : } \forall d' \in R, | d_{\text{timestamp}} - d'_{\text{timestamp}} | \leq R_{rvi} .$$

데이터베이스의 데이터를 시간적으로 올바르게 유지하려면 도착하는 자극이나 입력을 제때에 기록하고, 새로운 데이터를 제때에 계산하는 행위가 제때에 이루어져야한다. 데이터가 시간적으로 무효한 경우 수정 행위가 취해져야한다.

예) 배송 추적 시스템의 데이터베이스에 있는 배송차량의 위치, 속도, 방향 등은 반드시 최근의 것이어야 하고 시간적으로 상호연관 되어야 한다. 만일 시간적 유효성을 잃었을 때는 새로운 값을 구해 유효성을 재구성해야 한다.

한 데이터를 가지고 다른 데이터를 구하는 방법, 상대적 일관성 집합의 구성, 그리고 유도되는 데이터에 타임스탬프를 정하는 방법 사이에는 분명한 상호관계가 존재한다. 시스템에 과도

한 제약을 부가하지 않도록 하는 방법론적 접근이 필요하다. 즉 시간적 일관성 요구가 필요 이상으로 엄격해서는 안 된다. 이 문제는 중요한 것으로 시간적 일관성 요구는 행위에 대한 시간 제약이 되기 때문이다. 따라서 시간적 일관성 요구가 엄격할수록 행위에 대한 시간제약도 더 심해지고, 이것을 만족시키는 것은 더 어려워진다.

데이터의 avi와 rvi는 동적으로 변한다. 즉, 배송차량의 출발이나 도착하는 동안 속도와 위치는 자세히 감시되어야 한다. 즉 이들의 avi는 작아야 한다. 그러나 일단 출발한 주행시작 중일 때는 avi는 작아야한다. 그러나 일단 출발하여 주행 중 일 때는 avi를 증가 시켜도 된다. 환경의 상태를 더 이상 반영하지 못하는 데이터 역시 위치 추적 시스템이 유용하게 쓸 수 있다. 무엇보다도, 예측을 위해 쓸 수 있다. 예를 들면, 특정지역의 과거 주행속도는 미래의 경향을 판단하는데 쓰일 수 있다. 둘째로, 어떤 알고리즘은 적당한 상태 값을 가지고도 계산할 수 있다. 일반적으로 서로 다른 서브시스템들은 서로 다른 정도의 시간적 유효성을 필요로 한다.

### 3. 행위의 시간적 의미

행위가 시간적 제약 조건과 연관되는 이유를 요약하면 :

- 1) 시간 제약은 협력 시스템의 행위를 제한한다. 즉, 시스템에 도달하는 입력의 도착 율이나 시간을 제한 한다. 예) 배송 추적 제어 시스템에서, 운전사는 터미널로부터 30분 거리 이전에는 도착허가 및 하역작업을 요구 할 수 없다.
- 2) 시간제약은 시스템의 성능을 나타낸다. 즉 시스템이 입력에 대한 반응정도를 나타낸다. 예) 도착허가가 요청되었을 때, 응답은 30초 이내에 이루어져야 한다.
- 3) 시간적 일관성을 유지하도록 협력 시스템을 감시하고 시스템의 데이터베이스를 갱신하는 행위에 시간제약이 가해진다. 예) 자동차의 속도, 위치 등을 갱신하는 행위는 명시된 주기 동안에 이루어져야 한다. 이러한 행위는 현재 데이터가 얼마나 빨리 무효화되느냐에 따라 마감시간이 결정된다.

이러한 세 가지 시간 제약을 협력 사이의 관계를 지배한다. 첫 번째와 두 번째 시간 제약은 발생하는 입력 사건과 이에 대한 시스템의 응답이 관련된다. 이 응답 자체 역시 사건이 될 수 있다. 행위에 시간 제약을 직접 명시할 수 있다. 즉, 예로 “행위 A는 t시간 안에 완료되어야 한다”. 행위의 종료 역시 하나의 사건이므로 행위에 대한 시간 제약 역시 이 사건에 대한 시간 제약으로 볼 수 있다. 세 번째 경우는 협력 제어 시스템과 피 협력 제어 시스템을 동기화 시킬 때 주로 발생한다.

여러 가지 상황에서, 시간은 동기화 메커니즘으로 그리고 상호 조정을 위하여 사용된다. 따라서 정확성을 얻기 위한 시간 제약 조건의 집합을 최소화하고 덜 엄격하게 하는 것이 중요하다.



시간 제약조건을 사건, 데이터, 행위와 연관 지을 수 있다. 사건은 행위를 유발시키고 데이터의 시간적 일관성은 적기 행위를 통해 유지된다. 따라서 협력시스템의 관점에서 보면, 여러 가지 형태의 시간 제약조건은 행위가 시간제약을 받는 실행을 하도록 한다. 물론 데이터와 사건의 시간 제약 요구를 행위의 시간제약으로 변환하는 방법은 아직 연구해야 할 문제이다.

## V. ECA 규칙 구성

시간 제약의 적기 협력 시스템 설계자의 주 관심은 시간제약을 갖는 행위를 스케줄 하는 것인데, 이 행위가 지연되거나 전혀 실행되지 않았을 때의 벌칙을 최소화하고 이 행위로 인한 시스템의 이익은 최대화하는 것이다.

한 행위의 지연이나 미 실행으로 인한 벌칙이 아주 클 경우 이것을 안전-임계의(safety-critical) 또는 단호한 시간( hard time ) 제약 조건이라 한다. 예를 들면, 고객이 주문 취소를 할 수 있는 마감시간은 안전 임계 조건이다.

마감시간이 지난 후에 행위를 수행시 아무 값도 생기지 않고 또 이에 대한 벌칙이 없다면, 이것을 엄격한 마감 시간(firm deadline)이라 한다. 이러한 경우 대체행위가 가능하다. 예를 들면, 택배 직원이 물품을 수집 또는 배달하는 것은 정해진 일과 시간에 이루어져야 하므로 엄격한 마감 조건(firm deadline)을 갖는다. 이 마감 조건 내에 객체를 수집 또는 배달하지 못하면 다음 영업일에 다시 수집 및 배달하여야 하고 증가된 중요도에 의해 다시 수행이 시도되어야 한다. 이에 대한 ECA 규칙은 다음과 같이 표현할 수 있다:

Event : deadline of "object delivery";  
 Condition : action not completed;  
 Action : "try again with increased importance";

또 다른 예로는 고객이 물품을 한 번 주문 후에 주문 수정(삭제, 추가, 취소 등)을 하는 경우 허용된 시간 내에 작업을 끝내지 않으면 강제로 접속을 중지 시킨다. 이에 대한 규칙은 다음과 같다 :

Event : 5 minutes after " initiating modifying order";  
 Condition : steps not completed;  
 Action : within 1 minute "Abort modifying";

마감시간 후에도 실행될 수 있는 행위들이 많이 있는데, 이것은 행위의 결과를 사용하는 소비자가 비록 데이터가 늦게 얻어지더라도 상관없는 경우이다. 그러한 마감시간을 느슨한(soft)

제약이라 한다. 그러나 이 지연된 행위 다음에 다른 행위들이 있다면, 시스템은 전체 시간제약을 만족되도록 이 지연된 결과에 따라 잘 조절 되어야한다. 이에 대한 예는, 쇼핑 몰에서 고객이 주문한 물품을 각 공급업체로부터 수집하여 한 번에 배송하는 경우 수집된 물품정보는 데이터베이스에 저장되고 한데 모여 배송을 기다린다. 이 일련의 순서는 전체 마감시간을 갖는다. 이 전체 마감시간은 각 부분 행위의 마감시간으로 구성된다, 따라서 비록 일부 행위의 마감시간을 놓치더라도 전체 행위의 전체마감 시간을 만족하면 된다. 이 각 부분의 행위의 마감시간은 느슨한 조건이다.

엄격한(firm) 조건과 느슨한(soft) 시간제약 조건이 시스템에게 더 많은 융통성을 부여한다. 단호한 제약 조건(hard constraints)은 외부요건 때문에 생긴다. 따라서 그러한 요구의 수는 최소화하는 것이 중요하다.

엄격한 조건과 느슨한 시간 제약조건 때문에 시스템은 마감시간과 다른 파라미터를 조정하여 시간제약 위반시 조치를 취할 수 있는 메타 제어(meta-control) 알고리즘이 필요하다. 이러한 메타 제어 규칙 역시 ECA 규칙으로 표현될 수 있다.

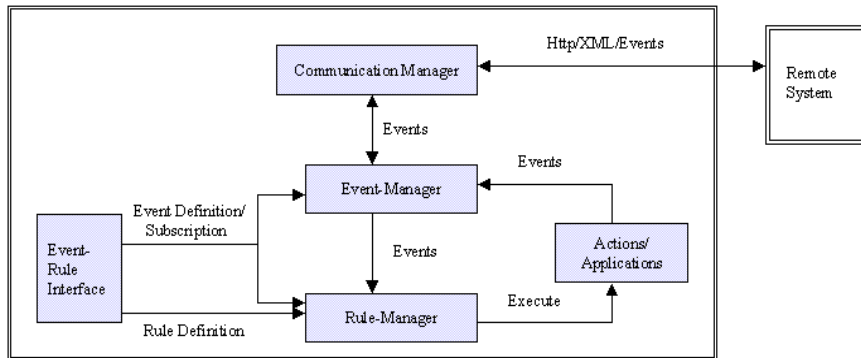
## VI. 프로토타입 구현과 평가

기업사이의 시간 제약이 필요한 적기 협력을 표현한 ECA 규칙 프로그램은 웹기반 능동 기능 컴포넌트에 의하여 실행되게 된다. 즉시 처리 협조를 요청하는 사건이 발생하면 이것이 탐지되어 이를 필요로 하는 규칙이 지역이면 지역 규칙 관리기에, 원격지 규칙이면 이를 필요로 하는 원격지 능동 기능 컴포넌트에게 전달한다. 이 장에서는 제안된 능동 기능 컴포넌트의 구성 요소와 프로토타입 구현을 설명한다.

### 1. 능동기능 컴포넌트

#### 가. 아키텍처

웹 기반 분산 기업 시스템 사이의 적기 협력을 표현한 ECA 규칙은 [그림 2]와 같은 능동 기능 컴포넌트에 의하여 처리된다. 능동 기능 컴포넌트는 통신 관리기(Communication\_Manager), 사건 관리기(Event\_Manager), 규칙 관리기(Rule\_Manager), 사건 규칙 인터페이스(Event\_Rule\_Interface), 행위/응용(Actions/Applications) 모듈로 이루어진다.



[그림 2] 능동 기능 컴포넌트 아키텍처

각 모듈의 중요 기능과 구조는 다음과 같다:

- 1) Communication\_Manager: HTTP 프로토콜을 이용하여 적기 협력을 위한 사건 메시지를 다른 서버의 통신 관리기와 주고받는 역할을 담당한다. 웹 서버의 Java Servlet으로 작성된 모듈로 2 가지 역할을 한다. 우선, 웹 서버를 통하여 외부에서 오는 XML 메시지에서 사건을 추출하여 사건 관리기에게 전달하는 역할(Send-Event)과 두 번째로, 내부 사건관리기가 전달하는 사건을 XML로 변환하여 HTTP post 명령으로 상대 능동 기능 컴포넌트의 통신 관리기에게 전달하는 역할(Receive-Event)이다.
- 2) Event\_Manager : 사건에 대한 정의 스키마를 관리하며 지역 사건과 원격지 사건을 인식하여 해당(구독하는) 규칙 관리기에게 전달하는 역할을 한다. 웹 서버의 Java Servlet으로 작성된 모듈로 필요한 사건 스키마 등록과 실행 시간 사건을 관리한다. 사건의 등록과 구독은 사건 규칙 인터페이스를 통하여 시스템 관리자나 프로그래머의 입력에 의하여 사건 관련 테이블에 사건에 관한 정의와 구독에 관한 사항을 등록한다. 그리고 실행 중에는 전달되어 오는 사건 사례를 Event-Comparator가 사건 관련 테이블의 subscriber-schema에서 확인하여 구독자가 지역인지 원격지인지를 결정하여 지역 규칙 관리기나 통신 관리기의 Send\_Event에게 파라미터와 함께 각각 전달한다. 사건 관련 테이블은 다음과 같이 4 개의 테이블로 구성된다.

event-schema=(event-name, no-of-parameters)

publisher-schema=(event-name, publisher)

subscriber-schema=(event-name, subscribers)

parameter-schema=(event-name, para-name, type, position)

이들 테이블은 데이터베이스에 저장된다. 따라서 사건 관리기는 JDBC를 이용하여 데이터베이스에 연결하여 이 테이블들을 관리한다.

- 3) Rule\_Manager : 사건 사례 테이블과 규칙 테이블을 관리하며 규칙을 평가 실행하는 역할을 한다. 하부 DBMS의 트리거를 이용하여 구현된 것으로 사건 사례 테이블(Event-Instance table) 과 규칙 테이블(Rule Table)을 갖고 있다. 그리고 해당 규칙의 평가 및 행위의 실행이 하부 DBMS에 의하여 이루어진다. 사건 규칙 인터페이스를 통하여 사건이 정의될 때 사건 정의에 의하여 해당 사건 사례 테이블을 형성한다. 규칙 역시 사건 규칙 인터페이스를 통하여 정의되고, 이 규칙들을 규칙 테이블에 저장하고 사건 사례와 내부 트리거로 연결된다. 사건 관리기로부터 전달되어 오는 사건 사례를 사건 사례 테이블에 저장 및 탐지를 하여 해당 규칙을 구동시킨다. 그리고 이 규칙의 조건을 평가한 다음 조건이 만족되면 규칙의 행위를 실행한다.
- 4) Event-Rule Interface : 시스템 관리자와 프로그래머를 위한 JSP로 작성된 인터페이스로 사건의 정의와 규칙의 정의 뿐 아니라 관리 즉, 검색 및 삭제와 수정을 할 수 있는 도구이다. 사건 관리기와 규칙 관리기의 기능을 이용하여 작업을 수행한다.
- 5) Actions/Applications은 DBMS의 API로 이루어진 내부 행위와 그 이외의 응용인 외부 행위를 의미한다. 즉시 협력을 요청하는 사건은 응용 관련 사건이므로 이를 발생시키는 포장 코드(wrapper code)가 생성되어야 하고 이를 사건 관리기에게 통보하는 raise-event() 함수 콜(function call)이 필요하다. ECA 규칙의 행위 부분은 즉시 협력을 위한 조치(협조 행위)를 취하는 부분이므로 앞의 규칙 관리기에서 설명한 것처럼 trigger body에 이에 대한 프로시저나 저장 프로시저 콜로 이루어진다. 그리고 이러한 행위(action)는 또 다른 사건을 발생시킬 수 있다.

## 나. 구성 요소 사이의 상호 작용

위 구성 요소 사이의 상호작용 과정을 build-time과 run-time으로 구분하여 정리하면 다음과 같다:

### 1) Build-time

1. 관리자나 프로그래머에 의한 사건 규칙 인터페이스를 통한 적기 협력 사건의 정의 입력
2. 사건의 정의는 사건 관리기에 의하여 사건 관련 스키마 테이블에 기록된다.  
사건을 발생하는 wrapper code 생성
3. 사건의 정의는 또한 규칙 관리기에게 전달되어 사건 사례 테이블이 생성된다.
4. 사건 규칙 인터페이스를 통한 사건 구독 입력
5. 사건 구독 정보는 규칙 관리기에 의하여 사건 관련 스키마 테이블에 기록
6. 사건 규칙 인터페이스를 통한 규칙의 정의
7. 규칙의 정의는 규칙 관리기에 의하여 규칙 테이블에 기록되고 해당 사건 사례 테이블에 대한 트리거를 생성하고 규칙의 행위 부분에 해당하는 코드 생성

## 2) Run-time

1. 지역 응용이나 행위에 의한 사건 발생이 사건 관리기에게 전달되거나 또는 원격지로부터의 사건이 통신 관리기를 통하여 사건 관리기에게 전달됨.
2. 사건 관리기는 사건 관련 스키마 테이블에서 해당 사건의 구독자를 확인
3. 구독자가 원격지 서버이면 통신 관리기에게 전달하고 지역이면 규칙 관리기에게 전달
4. 통신 관리기에게 전달된 사건은 파라미터와 프로토콜 포장기가 추가되어 원격지 통신 관리기에게 전달됨.
5. 규칙 관리기는 사건을 사건 사례 테이블에 삽입하여 트리거에 의한 해당 규칙을 구동시킨다.

## 2. 구현과 평가

제안된 웹 기반 능동 기능 컴포넌트의 프로토타입을 구현하기 위하여 설계시 고려 사항은 실용성, 데이터베이스와의 연동, 그리고 플랫폼 독립성이다. 따라서 구현 언어로는 Java, 상업용 DBMS로는 Oracle을 택하였고, Apache Tomcat Web Server, Xerces2 Java Parser, Windows2000 서버, Linux 서버를 사용하였다.

앞에서 언급한 능동 기능을 처음부터 모든 것을 설계하고 구현하거나, 기존 DBMS의 커널에 직접 융합시킨다면, 시간과 비용이 너무 많이 소모된다. 그리고 일반적으로 시스템의 안정성과 유지보수의 확실성을 고려하여, 현재 시스템 시장에서 널리 사용되고 있는 제품들을 사용하고 있다. 또 상업용 DBMS들은 최소한 기본적인 트리거 기능을 제공하므로 이를 이용하여 능동 기능 컴포넌트를 구축하면, 시스템 구축비용이 크게 절감되면서 기존 시스템을 그대로 적용할 수 있다. 본 연구에서는 상업용 DBMS 중에서 Oracle을 예로 하여 능동 기능을 구축하였다. 그러나 다른 제품들도 동일하게 적용될 수 있다.

적용된 예는 웹 기반 분산 환경에서 전형적인 B2B 전자상거래의 한 경우인 쇼핑몰과 공급자, 택배회사 사이의 적기 협력 구현에 적용하였다. 이들은 각자 자신의 사이트를 소유하고 있고 이를 통하여 B2B 전자상거래를 수행하고 있다. 인터넷상에는 다수의 쇼핑몰, 다수의 공급자, 다수의 택배회사가 있고 이들은 각각 독립적으로 영업 파트너(business partner)와 사업을 하고 있다. 적용의 편의성을 위하여 은행 부문은 제외하였다.

본 논문에서 제안하고 있는 웹 기반 능동 기능 컴포넌트는 시스템의 다른 부시스템(subsystem)과 느슨하게(loosely) 결합된 형태이므로 적기 협력 메커니즘을 필요로 하는 다른 어떤 시스템에도 쉽게 적용될 수 있다. 특히 Java Servlet과 상업용 데이터베이스 시스템의 기본적인 트리거를 이용한 능동 기능 컴포넌트 구현은 실용적이다. 즉, 대부분의 전자상거래 시스템에 채용되고 있는 웹 플랫폼에서 Java Servlet을 지원하고 있고 또 데이터베이스 시스템을 사용하고 있기 때문이다. 따라서 능동 기능이 데이터베이스와 융합되는 이점이 있어 기존의 상업용 데이터베이스 시스템의 능동 기능을 확장할 수 있다.

## VI. 결 론

이 논문에서 웹기반 B2B EC 분산 환경에서 기업 사이의 적기 협력을 위한 시간 제약 조건을 만드는 기원과 시간 제약조건의 의미를 사건, 데이터, 행위 관점에서 살펴보고, 이의 ECA 규칙에 의한 응용 구현 방법을 보였다. 이 방법에서는 일반적인 실시간(시간 제약) 관련 논문들에서처럼 모든 마감시간들이 사용자나 응용에 의해서 정의되지는 않는다.

유도된 마감시간의 융통성은 협력 시스템에 의해 충분히 사용되어야 한다. 마감시간이 유도되는 경우, 그 값은 적응적 방식으로 선택하여야 하고, 이 마감시간의 위반 역시 적응적으로 처리되어야 한다. 이르기 위해서는 시간제약 조건을 할당하고 유도하는 방법, 그리고 가장 덜 엄격한 값을 선택하는 방법이 필요하다. 시간 제약을 갖는 사건, 데이터, 행위가 있을 때 시스템은 시간 제약조건이 위배되었을 경우로부터 회복될 수 있어야하고 과부하 상태에서 성능 저하가 심하지 않아야 한다. 이러한 것들을 ECA 규칙으로 잘 표현할 수 있다.

앞으로의 연구로는 데이터나 사건의 시간 제약 요구에서 행위의 시간 제약으로의 변환 방법에 대한 연구와 주어진 제약 조건들에서 최적의 마감시간을 구하여 이를 규칙화하는 방법에 대한 연구 등이 필요하다.

## 참고 문헌

- 이동우(2005), B2B 전자상거래 환경에서 기업 사이의 적기 협력 지원을 위한 능동 기능 컴포넌트, 한국정보과학회논문지:컴퓨팅의 실제, vol. 11, no. 2, pp.165-179.
- Audsley, N., A. Burns, M. Richardson and A. Wellings(1991), Database Model for Hard Real-Time Systems, Technical Report, Real-Time Systems Group Univ of York UK. July.
- Dasarathy, B.(1985), Timing constraints of Realtime systems Constructs for Expressing Methods for Validating Them, IEEE Transactions on Software Engineering pp. 80-86, Jan.
- Medjahed, Brahim, et al.(2003), Business-to-business interations: issues and enabling technologies, VLDB journal, Springer-Verlag, April, pp.59-85.
- Paton, Norman W. and Oscar Diaz(1999), Active Database Systems, Computing Surveys, ACM.
- Purimetla, B., R. M. Sivasankaran, K. Ramamritham and J. A. Stankovic(1995), Real-Time Databases Issues and Applications, In Advances in Real-Time Systems, Sang Son Ed. Prentice-Hall.
- Ramamritham, K(1993), Real-Time Databases, International Journal of Distributed and Parallel Databases Vol. 1, No. 2.
- Son, Sang ed.(1995), Advances in Real-time Systems, Prentice-Hall.
- Widom, J. , S. ceri ed.(1996), Active Database Systems, Morgan Kaufmann.

**<Abstract>**

**Analysis of Timing Constraints for  
Timely Collaboration among Web-Based  
Distributed Business Systems**

**Tae-Jung Suh\* · Dong-Woo Lee\*\***

Timely collaboration among businesses is required these days such as in B2B E-Commerce. That is, emergency requests or urgent information among businesses should be processed in an immediate mode. In this paper we investigate the semantics on timing constraints of timely collaboration among businesses in terms of event, data, and action and present ECA rule construction related to the timing constraints and its violation. The prototype is implemented using web-based active functionality component.

**Key words : active database, timely collaboration among businesses, ECA rule, timing constraints semantics**

---

\* Professor, Division of Shop Master & Business, Woosong Information College

\*\* Correspondence, Professor of Department of Computer Science, Woosong University