u-Logistics 알림 시스템의 설계 및 구현[1)]

이용미[0] 김룡 서성보 오세원* 이용준* 류근호
충북대학교 데이터베이스연구실
*한국전자통신연구원
{ymlee[0], kimlyong, sbseo, khryu}@dblab.chungbuk.ac.kr
*{sewonoh, yjl}@etri.re.kr

Design and Implementation of the u-Logistics Alerting System

Yongmi Lee[0] Long Jin Sungbo Seo Se Won Oh* Yong Joon Lee* Keun Ho Ryu
Database Laboratory, Chungbuk National University
*Electronics and Telecommunications Research Institute

요 약

지능화된 사물이 스스로 공간의 상황을 인식하여 필요한 행위를 대신해 주는 유비쿼터스 컴퓨팅의 응용 환경하에서는 언제 어디서나 상황 정보를 분석하여 원하는 사용자에게 실시간으로 알려줄 수 있어야 한다. 그러나, 기존의 e-Logistics 물류 시스템에서는 실물의 흐름과 정보의 흐름간에 불일치가 존재하기 때문에 화물의 위치를 실시간으로 추적하는 것이 불가능하였다. 따라서, 우리는 이러한 문제들을 해결하기 위해 화물에 센싱 가능한 스마트 태그를 부착하고, 이로부터 실시간으로 수집된 정보를 이용하여 사용자에게 유용한 정보를 제공하는 u-Logistics 알림 시스템을 설계하고 구현하였다. 이 논문에서는 u-Logistics 시스템의 전체적인 구조를 기술하고, 특히 규칙의 정의에서부터 알림 서버가 실시간으로 이벤트를 수집하여 원하는 사용자에게 화물 또는 차량에 대한 정보를 제공하기까지의 알림 시스템의 전반적인 처리 과정을 기술한다. 마지막으로, 물류 환경에의 적용 시나리오를 통하여 u-Logistics 알림 시스템의 구현을 제시한다.

1. 서 론

오늘날의 컴퓨팅 환경은 물리공간에 존재하지 않는 사물을 지능화함으로써 사물이 스스로 공간의 상황을 인식하여 필요한 행위를 대신해 주는 유비쿼터스 환경을 기반으로 하고 있다. 이러한 환경하에서, 응용은 사물과 컴퓨터, 사람간의 기능적 연계를 실현시키기 위해 언제 어디서나 상황 정보를 분석하여 원하는 사용자에게 실시간으로 알려줄 수 있어야 한다.

이러한 요구는 물류 시스템에서도 예외는 아니다. 예를 들면, 사용자는 자신이 구매한 화물이 현재 어디쯤 이동중인지를 가능하면 정확하게 알고 싶어할 것이다. 그러나 기존의 물류 시스템으로 이러한 사용자의 요구를 실현하기에는 한계가 있다. 이는 물류 시스템의 운용자가 수작업으로 정보를 입력하는 환경하에서는 실물의 흐름과 그에 대한 정보가 데이터베이스에 저장되는 시간 사이에 불일치가 존재하기 때문이다. 이러한 정보들을 바탕으로 화물의 위치를 실시간으로 추적한다는 것은 불가능한 일이다.

이러한 문제들을 해결하기 위해, u-Logistics 시스템에서는 다음의 두가지 측면을 고려한 실시간 알림 시스템을 설계하고 구현하였다. 첫째, 모든 화물에 상품의 정보를 저장한 센싱 가능한 스마트 태그(Smart Tag)를 부착하고, 인터넷을 통하여 이들로부터 센싱되는 물류 정보를 실시간으로 수집한다. 둘째, 이렇게 수집된 물류 정보를 기반으로 화물의 위치를 실시간으로 추적하거나, 사용자에게 보다 유용한 물류 정보를 제공한다.

이 논문에서 소개하는 알림 시스템의 상세한 구조 및 모델은 [1]의 설계를 바탕으로 한다. 특히, 위의 두 가지 측면 중에서 스마트 태그에 관한 연구는 [2]를 기반으로 한 공동 선행연구로써 진행되었으므로, 이 논문에서는 센서 네트워크를 위한 인프라가 구축되어 있다고 가정하고 상세한 기술은 생략한다.

이 논문의 구성은 먼저 u-Logistics 시스템의 전체적인 구조를 개략적으로 설명한다. 그리고 알림 시스템의 기능을 크게 규칙 관리기(Rule Manager)와 알림 서버(Alert Server)로 구분하여 이벤트의 분석과 규칙의 비교 방법등을 상세히 기술한다. 마지막으로, 물류 환경에의 적용 시나리오를 통하여 u-Logistics 알림 시스템의 구현을 제시한다.

2. u-Logistics 시스템의 구조

u-Logistics 시스템의 개략적인 구조를 설명하고 있는 그림 1은 화물의 발송 준비로부터 고객에게 배달되기까지 컴퍼넌트간의 연계를 보여주고 있다.

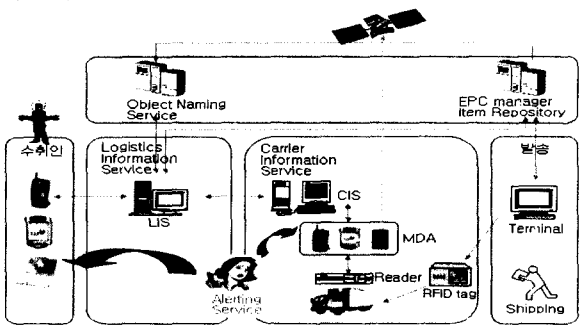

그림 1. u-Logistics 시스템의 구조도

u-Logistics 시스템에서는 모든 단위 화물에 상품에 대한 고

1) 이 논문은 한국전자통신연구원 우정기술연구센터의 "e-Logistics 통합 플랫폼 및 지능화 시스템 개발" 과제의 지원에 의해 수행되었음

유한 정보를 저장하는 RFID Tag가 부착된다. 이 Tag가 차량에 부착된 RFID Reader를 통과함으로써 화물의 선적 및 하역과 관련된 정보들이 자동으로 인식된다. 또한, 차량에 탑재된 MDA(Mobile Digital Assistant)는 차량의 출발과 도착 및 현재 이동중인 위치에 관한 정보들을 주기적으로 운송업체의 CIS에 전송한다.

이 논문에서는 이와 같이 센서로부터, 예를 들면, RFID Tag나 GPS로부터 끊임없이 수집되는 물류 정보들을 이벤트라고 정의한다. 알림 시스템은 이러한 이벤트들을 이용하여 운송중인 화물의 위치를 추적하거나, 사용자가 원하는 물류 정보를 실시간으로 제공해 준다.

3. 알림 시스템

u-Logistics에서의 알림 시스템의 기능은 크게 규칙 관리기(Rule Manager)와 알림 서버(Alert Server)로 구분할 수 있다. 이 장에서는 규칙의 정의에서부터 이벤트의 분석, 규칙과의 비교를 통해 정보를 통지하기까지의 과정을 상세히 기술한다.

3.1. 규칙 관리기(Rule Manager)

규칙이란 사용자가 자신의 화물 또는 차량에 대해 관심있고, 통지 받기를 원하는 정보를 정의한 것을 의미한다. 사용자는 규칙을 정의함으로써 Alert Server로부터 관련된 정보를 제공 받을 수 있다.

u-Logistics에서는 규칙의 관리를 위해 Rule Manager라는 도구를 제공한다. Rule Manager는 [3]이 하나의 속성에 대한 정의가 끝나면 다음의 속성을 보여주는 단계적인 인터페이스를 제공하는 것과는 달리 사용자에게 그래픽 인터페이스를 제공한다. 따라서, 사용자는 액터와 물류 객체들을 이용하여 좀 더 친숙하고 편리하게 규칙들을 정의할 수 있다. 현재 Rule Manager에서는 아래와 같은 규칙들을 정의할 수 있다.

R1: 운송차량이 정해진 거점 통과 시각을 120분이상 초과했을 경우 알려 주세요.
R2: 나의 화물의 위치를 30분마다 알려주세요.
R3: 운송 차량이 특정 지점을 통과하면 알려주세요.
R4: 운송 차량이 특정 화물에 대한 운송을 시작하였음을 알려주세요.
R5: 나의 화물이 도착지 반경 10Km 이내로 진입하면 알려주세요
R6: 운송 차량이 종점으로 복귀하면 알려 주세요.

또한, 이렇게 정의된 규칙은 XML 형식으로 데이터베이스에 저장되어 규칙의 관리가 용이하도록 지원한다.

3.2. 알림 서버(Alert Server)

Alert Sever의 역할은 크게 이벤트의 분석, 사전에 정의된 규칙과의 비교 및 정보의 통지로 구분할 수 있다.

3.2.1. 이벤트의 분석

Alert Server가 실시간으로 수집한 이벤트를 분석하는 단계이다. 우리는 u-Logistics 시스템에서 이벤트의 발생을 표 1과 같이 차량의 운송 시작과 종료, 차량의 이동, 화물의 선적과 하역의 5가지로 경우로 정의하였다.

아래의 그림 2는 Alert Server가 수신한 이동중인 차량의 주기적인 위치 보고시 발생한 이벤트의 예를 보여주고 있다. 이들 이벤트들은 XML 형식으로 표현되며, 하나의 이벤트는 메타 데이터가 기술되는 <Header>와 이벤트에 대한 자세한 내용이 기술되는 <Body>로 구성된다.

표 1. 이벤트의 종류

이벤트	설명
MDA_REGISTERED	MDA의 설정사항을 CIS에 등록
MDA_TRANSPORT_STARTED	차량의 운송 시작시 발생
MDA_LOCATION_UPDATED	차량의 주기적인 위치 보고
MDA_ITEM_UNLOADED	화물의 하역시 발생
MDA_TRANSPORT_ENDED	차량의 운송 종료시 발생

```
<?xml version="1.0" encoding="EUC-KR"?>
<CISEvent type="MDA_LOCATION_UPDATED">
  <Header>
    <Source id = "CIS_001" />
    <Destination host="#ALERT_HOST_NAME" port="3000" />
    <Timestamp>2004-01-19T01:29:29Z</Timestamp>
  </Header>
  <Body>
    <MDA id="MDA-1075707079" />
    <LocationInfo>
      :
      <Latitude>
        :
      </Latitude>
      <Longitude>
        :
      </Longitude>
      <Velocity>33.1</Velocity>
      <Course>120.1</Course>
    </LocationInfo>
  </Body>
</CISEvent>
```

그림 2. 주기적인 위치 보고 이벤트의 예

Alert Server는 XML 스트링으로부터 'CIS_001'라는 CIS가 'MDA-1075707079'라는 MDA를 탑재한 차량으로부터의 받은 이벤트로써, 이동중인 차량의 현재 위치, 속도 및 진행 방향등이 포함되어 있다고 분석할 것이다. 그리고, 이벤트는 유형에 따라 규칙과의 비교에 이용되거나 데이터베이스에 저장(append-only)되어 로그 및 통계 분석을 위한 자료로써 활용한다.

3.2.2. 규칙의 비교(Rule Matching) 및 정보의 통지

이 단계에서 Alert Server는 사용자에게 물류 정보를 제공하기 위한 입력으로써 분석된 이벤트를 필요로 하는 규칙이 있는지를 확인한다. 그림 3은 Alert Server가 이동중인 차량에 대한 주기적인 이벤트를 분석한 후 규칙이 존재하는지를 비교하는 과정을 간략하게 기술한 알고리즘이다. 이 알고리즘은 이벤트에 기술된 MDA_ID, 발생 시간, 위치값 들을 입력으로 하며, 출력으로는 메시지를 전송했는지 여부를 반환한다.

u-Logistics 시스템에서 모든 차량은 매일 아침에 미리 정해진 배달 일정(Schedule_DB)에 따라 화물을 싣는다고 가정한다. 이 때, Alert Server는 화물 정보가 존재하지 않는 이벤트로부터 운송중인 화물에 대한 정보를 찾기 위해 차량의 스케줄(Schedule_DB)을 이용하여, 각 화물에 대해 정의된 규칙(RuleSet, Profile)들이 존재하는가를 검사한다.

Alert Server는 해당 규칙에 대해 마지막으로 통지한 시간과 이벤트가 발생한 시간 간격이 사용자가 규칙으로 정의한 시간

간격을 초과할 때, 통지를 위한 메시지를 생성한다. 예를 들면, '당신의 화물 ○○이 9:00 현재 ○○지점을 통과하였습니다'. 이 과정에서 이벤트로부터 송신한 위도와 경도값을 지명으로 변환하는 작업을 필요로 한다. 이때 메시지의 통지는 사용자가 정의한 규칙에 따라 SMS 메시지 또는 이메일로 전송한다.

```
1  Input  : MDA_ID  : MDA identifier
2           date    : 이벤트가 발생한 시간
3           latitude  : 위도
4           longitude : 경도
5  Output : boolean value : 메시지의 송신 여부(true, false)
6  Begin
7    if ({MDA_ID, date} ∈ Schedule_DB) { // 배달정보
8      select EPC_list
9      for (each EPC ∈ EPC_list) do {
10       if (EPC ∈ RuleSet_DB) {  // 규칙에 대한 정보
11         select ruleNo_list
12         for (each ruleNo ∈ ruleNo_list) do {
13           compute interval //마지막 통지와의 시간 간격
14           if (interval <= interval of RuleSet) return false
15           if (ruleNo ∈ Profile_DB) { //수신자 정보
16             select profile_list
17             for (each profile ∈ profile_list) do {
18               translate (latitude,longitude)
                       to the name of place //지명변환
19               generate the message
20               if method = 1
21                 send SMS
22               else
23                 send e-mail
24             } // end of for each profile
25             return true
26           } // end of if (ruleNo ∈ Profile_DB)
27         } // end of for each ruleNo
28       } // end of if (EPC ∈ RuleSet_DB)
29     } // end of for each EPC
30   } // end of if ({MDA_ID, date} ∈ Schedule_DB)
31 return false
32 End
```

그림 3. 주기적 위치 보고 이벤트에 대한 매칭 알고리즘

4. 적용 시나리오 및 구현

이 장에서는 u-Logistics 알림 시스템에서 Rule Manager와 Alert Server의 구현을 이벤트의 발생으로부터 사용자가 원하는 정보를 통지하기까지 간단한 시나리오를 통해 설명한다.

청주시 개신동에 물류 창고를 둔 '12가 3456' 차량의 2004년 2월 25일 운송 스케줄은 아래와 같다. 또한, MDA가 이동중인 차량의 위치를 보고하기 위한 주기를 10분으로 설정하였다.

화물선적(09:30)→창고 출발(10:00)→분평동(10:50,EPC_001)→용암동(11:30,EPC_002)→우암동(13:20,EPC_003)→운천동(13:50,EPC_004)→봉명동(14:30,EPC_005)→차고도착(15:10)

화물의 구매인인 홍길동은 자신이 구매한 화물이 운송되기 전에 "나의 화물 EPC_001이 이동중인 위치를 20분마다 홍길동 자신에게는 SMS 메시지로, 화물의 수취인인 이몽룡에게는 E-mail로 알려주세요" 라는 규칙을 정의하였다. 그림 4는 구현된 Rule Manager를 이용하여 이 규칙을 정의한 결과 화면이다.

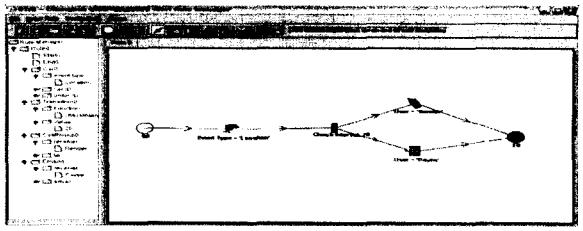

그림 4. Rule Manager를 이용한 규칙의 정의

9:30분 운송 스케줄대로 모든 화물을 실은 차량은 10:00에 운송을 시작하였다. 차량이 출발 후 10:20분에 수곡동을 지나면서 차량의 위치 이벤트가 발생하였다. 이에 대해 Alert Server는 EPC_001에 대해 출발 후 발생한 위치 이벤트와의 시간 간격이 위에서 정의한 규칙의 20분과 매칭되므로, '화물 EPC_001이 10:20분 현재 수곡동을 통과하였습니다.' 라는 메시지를 생성한다. 그리고 이 메시지를 홍길동에게는 SMS로, 이몽룡에게는 E-mail로 전송한다.

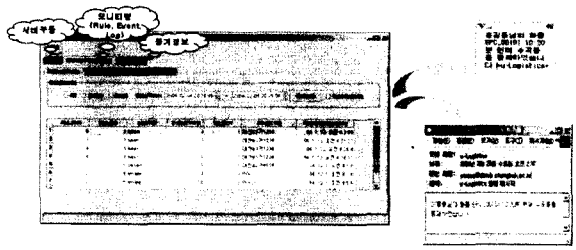

그림 5. Alert Server의 모니터링 툴과 알림 메시지의 통지

그림 5의 오른쪽은 구현된 u-Logistics 알림 시스템에 의하여 위에서 생성된 메시지가 SMS와 E-mail로 전송된 결과이다. 또한, 왼쪽은 관리자가 정의된 규칙, 발생한 이벤트, 통지 결과등을 확인할 수 있는 모니터링 툴을 보여주는 화면이다.

5. 결 론

지금까지 우리는 화물의 위치를 실시간으로 추적하기 위해 기존의 물류 시스템에 알림 시스템을 적용한 u-Logistics 알림 시스템에 대해 기술하였다. 먼저 전체 시스템의 구조를 개략적으로 설명하였고, 그 중 알림시스템의 기능을 규칙 관리기(Rule Manager)와 알림 서버(Alert Server)로 구분하여 규칙의 설정, 이벤트의 분석, 이벤트와 규칙과의 비교 및 정보의 통지를 기술하였다. 또한, 시나리오를 통하여 u-Logistics 알림 시스템의 구현을 제시하였으며, 실제 환경에 실험 적용하였다.

참 고 문 헌

[1] 이용미, 서성보, 박주상, 이용준, 류근호, u-Logistics 시스템에 적용가능한 알림 시스템 모델, 한국정보과학회, pp 112~114, 2003.

[2] Duncan McFarlane, Auto-ID Based Control - An Overview, White Paper Number CAM-AUTOID-WH-004, Auto-id Center Institute for Manufacturing, Cambridge, 2002. http://www.autoidcenter.org/abstractTest.asp?ID=20

[3] Amit - Active Middleware Technology Overview, IBM Research Lan in Haifa, 2002. http://www.haifa.il.ibm.com/projects/software/amit/index.html