

USN 환경에서의 상황정보 분석 시스템의 설계 및 구현[†]

김성호*, 이용미*, 남광우**, 이준욱***, 류근호*

*충북대학교 데이터베이스 연구실

**군산대학교 컴퓨터정보공학과

***전자통신연구원

*{kimsungho, ymlee, khryu}@dblab.chungbuk.ac.kr

** kwnam@kunsan.ac.kr

*** junux@etri.re.kr

Design and Implementation of Contextual Information Analysis System in USN Environment

Cheng Hao Jin*, Yongmi Lee*, Kwang Woo Nam**, Jun Wook Lee***, Keun Ho Ryu*

*Database/Bioinformatics Laboratory, Chungbuk National University, Korea

**Dept. of Computer Information Science, Kunsan National University, Korea

***Electronics and Telecommunications Research Institute, Korea

요 약

최근 IT 기술의 발전과 더불어 다양한 종류의 컴퓨터가 사람, 사물, 환경 속으로 스며들어 네트워크 망을 이루는 USN(Ubiquitous Sensor Network) 환경이 급속히 보급되고 있다. USN 환경에서 수집되는 센서 스트림 데이터는 연속적이며 동적으로 끊임없이 전달이 되기 때문에 그 양이 방대하며 또한 제한된 메모리를 사용하기 때문에 모든 데이터를 저장하여 처리할 수는 없게 된다. 스트림 데이터의 이러한 특성 때문에 본 논문에서는 입력되는 스트림 상황정보에 대해서 신속한 상황 분석 서비스를 진행하기 위하여 슬라이딩 윈도우 기법을 지원하는 상황정보 분석 시스템을 제안한다. 이 시스템은 온도, 습도, 조도 등 스트림 데이터에 대해서 WHEN-DO 상황질의모형을 적용하여 상황질의모형의 조건 만족 여부를 판단하고 특정 행동을 취한다. 따라서 본 논문에서 제안한 시스템은 실시간 건물의 상황정보를 수집하여 상태를 모니터링 하는 등 많은 USN 응용분야에 적용이 가능하다.

1. 서론

최근 IT 기술의 발전과 더불어 다양한 종류의 컴퓨터가 사람, 사물, 환경 속으로 스며들고 이들이 네트워크 망을 이루는 USN(Ubiquitous Sensor Network) [1, 2, 3] 환경이 급속히 보급되고 있다. USN 환경이란 모든 사물에 통신기능이 있는 스마트 전자 태그 및 센서를 부착하고 사물의 인식정보와 주변의 환경정보를 수집하여 언제 어디서나 상응한 서비스를 제공하는 환경을 가리킨다. 이러한 USN 환경은 물류, 의료, 통신, 가전 등 응용분야에 광범위하게 응용이 되며 인간 생활의 편리성과 안전성을 도모한다. USN 응용분야 중에서 가장 크고 가장 중요한 위치를 차지하고 있는 상황분석 서비스는 다양한 센서들로부터 다양한 종류의 상황정보의 수집을 통해 인식하고 사용자에게 상황에 맞는 서비스를 제공한다. 그러나 USN 환경에서 수집되는 스트림 데이터는 속도가 아주 빠르고 끊임

없이 지속적으로 입력이 된다. 이러한 스트림 데이터 형식으로 입력되는 상황정보에 대해서 정확하고도 신속한 상황분석 서비스를 진행하려면 빠른 데이터 처리가 필요하다. 따라서 본 논문에서는 USN 환경에서 수집되는 스트림 상황정보 데이터에 대해서 슬라이딩 윈도우 기법을 지원하는 WHEN-DO 상황질의모형을 적용하여 신속하고도 정확한 상황정보 분석을 진행할 수 있는 상황정보 분석 시스템을 제안한다.

전체적인 논문의 구성은 다음과 같다. 2 장에서는 스트림 데이터에 대한 기존의 상황분석 시스템을 살펴보고 3 장에서는 이 시스템에서 적용할 WHEN-DO 상황질의모형에 대해 설명한다. 4 장에서는 상황정보 분석 시스템 구조도와 알고리즘, 5 장에서는 시스템 구현에 대해서 기술한다. 마지막으로 6 장에서는 논문에 제안한 시스템에 대한 결론을 맺는다.

2. 관련 연구

[†] 이 연구는 교육과학기술부와 한국산업기술재단의 지역혁신인력양성사업과 MKE/IITA의 IT R&D 과제 [2006-S-022-01, RFID/USN 미들웨어 플랫폼 기술 개발]로 수행된 연구결과임

U-healthcare 분야에서 SAPhyRA (Stream Analysis for Physiological Risk Assessment) [4]은 이력데이터로부터 정상적인 행동과 이상한 행동을 판단하고 이로부터 위험수준 모델을 만들고 다음 슬라이딩 윈도우 기법으로 모든 가능한 생리학 스트림 정보를 분석하여 [4]의 공식에 의해서 환자의 위험수준을 측정한다. 분석하여 얻어낸 값이 크게 되면 환자의 위험성이 높고 값이 작으면 위험성이 낮은 것으로 간주한다. 여기에서 타임 윈도우 크기는 아주 중요한 역할을 일으키며 윈도우 크기가 작게 되면 위험성 발견은 아주 순간적이며 전에 모든 값들은 모두 무시하게 되지만 윈도우 크기가 크다면 실시간으로 수집되는 긴 시간 동안 이상한 행동으로 인해 수집되는 생리적 스트림 데이터로부터 위험수준을 정확하게 판단할 수 있다. 그리하여 위험 수준이 일정한 수치를 초과하게 되면 의사들은 신속하게 환자에게 치료방안을 낼 수 있다. TelegraphCQ [5]는 미국 버클리 대학에서 연속 질의를 사용하여 대용량의 스트림 데이터로부터 원하는 정보를 얻을 수 있도록 지원하기 위하여 PostgreSQL 기반으로 만들어졌다. 또한 슬라이딩 윈도우 연산을 위하여 WindowIS 연산자를 제공한다.

3. WHEN-DO 상황질의모델

상황질의모델은 상황분석 서비스에서 이슈로 등장하고 있으며 잘 설계된 상황질의모델은 상황인식 서비스 결과에 아주 중요한 영향을 준다. 이 부분에서는 우리가 제안하는 공간 상황정보 분석 시스템에서 적용하게 될 WHEN-DO 상황질의모델을 그림 1에서 소개한다.

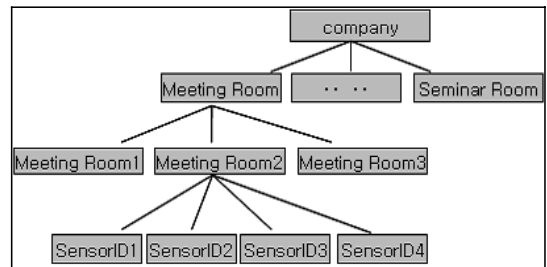
```

ContextModel ::= 'WHEN' <ContextCondition> 'DO' <ContextAction>
ContextCondition ::= <ContextExpression> [ <WindowExpression> ]
ContextExpression ::= '<CONTEXT>' ( ' <ObjectName> ' ; <ContextType> ; ; <Observation> ' )
ContextType ::= ( <AggregateOperator> ' ( ' <TypeName> ' ) ) | <TypeName> | 'ACTIVITY'
AggregateOperator ::= 'SUM' | 'AVG' | 'MIN' | 'MAX' | 'COUNT'
Observation ::= ( <NonStateOperator> ; ; <Literal> ) | ( <StateOperator> ; ; <State> )
NonStateOperator ::= <RelationalOperator> | <VariationOperator>
RelationalOperator ::= '>' | '<' | '=' | '<' | '>' | '=' | '<' | '>'
StateOperator ::= 'IS' | 'ISNOT'
VariationOperator ::= 'RISING' | 'FALLING'
State ::= 'USING' | 'RUNNING'
WindowExpression ::= ' ( ' 'RANGE' <IntegerLiteral> <TimeUnit> 'SLIDE' <IntegerLiteral> <TimeUnit> ' )'
TimeUnit ::= 'HOURS' | 'HOUR' | 'MINUTES' | 'MINUTE' | 'SECONDS' | 'SECOND'
ContextOperator ::= 'AND' | 'OR'
ContextAction ::= <ActuateExpression> { { <ContextOperator> <ActuateExpression> }
ActuateExpression ::= 'ACTUATE' ( ' <JavaLib> <ActuateCommand> ' )
ActuateCommand ::= 'ON' | 'OFF'
    
```

(그림 1) WHEN-DO 상황질의모델

이 상황질의모델은 크게 WHEN 조건절과 DO 구동절로 이루어진다. 즉 WHEN 조건을 만족할 때에 DO 부분의 행동을 취한다는 목적에 의해서 디자인 되었다. WHEN 조건절은 ContextExpression 과 WindowExpression 으로 구성되고 ContextExpression 은 또 ObjectName, ContextType, Observation 로 이루어진다.

ObjectName 은 사용자가 관찰하려고 하는 장소 혹은 한 개의 센서를 가리키며 ObjectName 이 장소일 때에는 이것은 포함관계를 가질 수 있다. 한 회사의 구조가 그림 2 와 같이 되어있고 각 방마다 한 개 혹은 한 개 이상의 센서가 설치 되어 있다고 가정한다. ObjectName 을 회의실로 선택하게 되면 이때에는 회의실 1, 2, 3 을 모두 포함하게 되고 만약 전체 ObjectName 을 전체 회사로 선택한다면 회사 내에 있는 모든 방을 다 포함하게 된다. ContextType 은 센서가 수집하는 스트림 상황정보의 타입 즉 예를 들어 센서가 온도, 습도, 조도 데이터를 수집한다면 ContextType 은 온도, 습도, 조도 가 된다. Observation 은 수학에서의 비교연산자 혹은 SQL 언어에서 쓰는 집계연산자, 혹은 수치의 변화 상태가 상승인지 하강인지를 나타낼 때에 사용이 된다.



(그림 2) 회사 구조

WindowExpression 에 속하는 range 와 slide 두 항목은 스트림 데이터를 처리할 때 사용된다. 스트림 데이터는 양이 크고 연속적으로 입력이 되기 때문에 한번에 처리할 수 없고 한번에 한 개 단위의 슬라이딩 윈도우만큼 한 데이터를 처리해야 한다. Range 는 슬라이딩 윈도우의 크기, slide 는 슬라이딩 윈도우가 움직이는 주기를 가리킨다. TimeUnit 은 슬라이딩 윈도우의 시간 단위를 표시한다. 마지막으로 DO 구동절은 javaLib 와 ActuateCommand 두 항목으로 구성되어 있으며 javaLib 는 기계의 종류를 나타내고 ActuateCommand 는 이 기계의 작동상태 즉 켜거나 끄는 상태를 나타낸다. 표 1 은 WHEN-DO 상황질의모델에서 각 항목이 가질 수 있는 값을 열거 한 것이다.

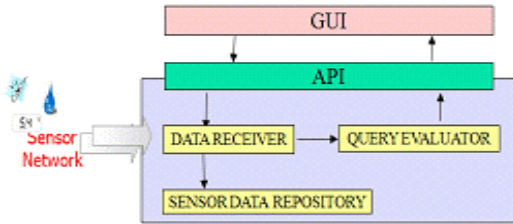
표 1 WHEN-DO 상황질의모델의 항목이 가지는 값

ELEMENT	VALUE
ObjectName	SensorID, Room, Rooms
ContextType	Temperature, Light, Humidity, Voltage, ACTIVITY, aggregate operation
AggregateOperator	SUM, AVG, MIN, MAX, COUNT
RelationalOperator	>, <, =, <=, >=
VariationOperator	RISING, FALLING
StateOperator	IS, NOT
State	USING, RUNNING
TimeUnit	HOURS, HOUR, MINUTES, MINUTE, SECONDS, SECOND
JavaLib	Air condition, Light and other things
ActuateCommand	ON, OFF

4. 상황정보 분석 시스템

이 부분에서는 연속적으로 끊임없이 입력되는 스트림 데이터에 대해서 신속하게 상황분석 서비스를 진행하기 위한 슬라이딩 윈도우 기법을 지원하는 공간 상황정보 분석 시스템을 소개한다. 우리가 제안하는 시스템 구조도는 그림 3 과 같다.

이 시스템은 GUI(Graphic User Interface), API(Application Program Interface), Data Receiver, Query Evaluator 와 Sensor Data Repository 로 구성이 된다. GUI에서는 사용자로부터 WHEN-DO 상황질의모형을 생성하고 사용자 요구질의를 만들어낸다. Data Receiver 는 WHEN-DO 상황질의모형에서 지정한 ObjectName 으로부터 데이터를 받고 GUI 로부터 API 를 통해 받은 WHEN-DO 상황질의모형을 적용한다. Query Evaluator 에서는 상황질의모형 적용하여 얻은 값이 상황질의모형의 조건 만족여부를 판단하고 조건을 만족하게 되면 WHEN-DO 상황질의모형에서의 DO 구동질의 행동을 취하게 된다. Sensor Data Repository 는 센서 ID, 센서의 위치정보와 부분적인 센서 데이터 등을 저장한다.



(그림 3) 상황정보 분석 시스템

설명: 시스템 구동 알고리즘

1. GUI로부터 상황질의모형을 만든다.
2. WHEN-DO상황질의모형의 objectName에 속하는 센서 집합을 구한다
3. 절차 2에서 얻어낸 센서가 수집한 데이터를 저장한다.
4. WHEN-DO상황질의모형을 적용하여 상황질의모형의 조건 만족 여부를 판단한다.
5. 조건을 만족하면 WHEN-DO상황질의모형의 구동질의 행동을 취한다.
6. 슬라이딩 윈도우를 움직인다.

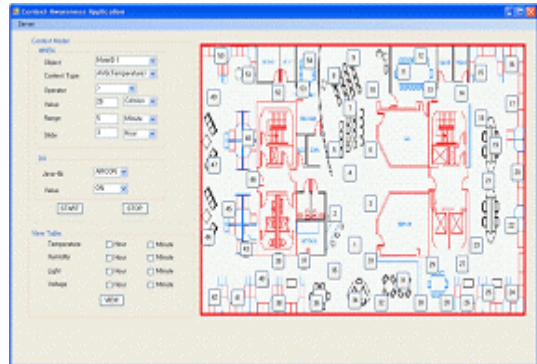
(그림 4) 상황정보 분석 시스템 구동 알고리즘

그림 4 는 시스템의 구동을 위해 사용되는 알고리즘에 대해서 설명한다. 이 알고리즘에 대해 상세하게 설명하면 다음과 같다. 절차 1 에서는 GUI 로부터 WHEN-DO 상황질의모형의 모든 항목을 설정하고 절차 2 에서는 WHEN-DO 상황질의모형의 한 개 항목인 ObjectName 에 속하는 센서들의 집합을 구한다. 회사의 구조가 그림 2 처럼 되어있다고 가정한다. 회의실을 ObjectName 으로 선택하였다면 우선 회의실에 포

함되어 있는 모든 회의실 3 개를 찾아내고 다음 3 개 회의실에 있는 모든 센서들의 정보를 알아내야 한다. 그렇지 않고 단순히 회의실 1 을 ObjectName 으로 선택하였다면 다만 회의실 1 에 속해있는 센서 정보만 알아내면 된다. 절차 3 에서는 절차 2 에서 얻은 센서들로부터 수집된 센서 데이터를 저장한다. 절차 4 에서는 절차 3 으로부터 수집된 스트림 데이터에 WHEN-DO 상황질의모형을 적용하여 스트림 데이터에 대한 질의결과가 WHEN-DO 상황질의모형의 WHEN 조건 만족 여부를 판단하고 조건을 만족하게 되면 절차 5 에서 WHEN-DO 상황질의모형의 DO 부분의 행동을 취한다. 절차 6 에서는 한 개 시간 단위가 지날 때마다 슬라이딩 윈도우를 움직인다.

5. 구현

그림 5 는 우리가 제안하는 상황정보 분석 시스템의 구현 화면이다. 시스템 구현 환경으로는 CPU Mobile DualCore Intel Merom 1800MHz, 1G DDR2 SDRAM 메모리와 Window XP Professional 운영체제를 사용하였다. 사용한 프로그래밍 언어로는 .Net Framework SDK v2.0 C#을 사용하였고 사용한 개발도구로는 Visual Studio 2005 이다. 구현에서 사용한 데이터셋으로는 Intel Lab 센서 데이터 [6]이다.



(그림 5) 공간 상황정보 분석 시스템 화면

이 데이터는 온도, 습도, 조도, 배터리 잔여 량 등으로 구성이 되었는데 구현에서는 배터리 잔여 량을 사용하지 않고 센서로부터 온도, 습도, 조도만이 스트림 데이터로 입력이 된다고 가정한다.

그림 5 에서 보는 바와 같이 시스템 구현화면은 주로 세 개 부분으로 이루어졌다. 화면의 왼쪽 상단에 WHEN-DO 상황질의모형, 그 아래 부분에 데이터보기, 그리고 오른쪽 부분에 센서가 회사 전체에 분포된 배치도를 볼 수 있다. 구현화면의 WHEN-DO 상황질의모형부분에서는 3 장에서 설명했던 WHEN-DO 상황질의모형의 모든 항목을 설정할 수 있다. 상황질의모형의 모든 설정을 마치고 START 버튼을 클릭하게 되면 클라이언트는 ObjectName 에 속해 있는 센서들로부터 수집되는 데이터에 WHEN-DO 상황질의모형을 적용하여 WHEN 조건을 만족하게 되면 그 DO 구동질의 행

동을 취하여 상응한 변화를 오른쪽 센서 분포 배치도에 나타내게 된다.

계속하여 그림 2 의 회사구조를 사용한다고 가정한다. 회의실을 ObjectName 으로 선택하였을 때 가정하여 회사 3 개 회의실의 5 분 단위의 평균 온도가 매 3 분마다 지속적으로 상승하게 된다면 이 회의실에는 화재가 발생 할 가능성이 상당히 높다. 이러한 경우에 오른쪽 센서 배치도에서 3 개 회의실에 분포된 센서의 색상을 빨간 색으로 변경시키면서 화재의 위험성을 알리고 화재의 발생을 방지할 수 있다. 데이터를 보기는 사용자가 원하는 상황정보 타입을 선택한 시간 단위 내에서의 평균값, 최소치, 최대치 값을 볼 수 있다. 또한 오른쪽 부분의 각 센서를 클릭하게 되면 그 센서의 센서 ID, 그 센서가 속해있는 장소, 센서의 좌표, 수집하는 데이터의 값을 보여준다.

6. 결론

최근 IT 기술의 발달로 인해 필요한 곳에 센서를 부착하여 주위의 상황정보(온도, 습도, 오염정보, 균열 정보 등)를 수집하는 USN 환경이 가능해졌다. USN 환경에서 수집되는 센서 데이터는 속도가 아주 빠르고 끊임없이 연속적으로 입력이 된다. 스트림 데이터 그 자체의 특성 때문에 모든 데이터를 저장할 수는 없게 된다. 따라서 본 논문에서는 온도, 습도, 조도 등 무한히 연속적으로 입력되는 상황정보 스트림 데이터에 대해서 신속한 상황분석 서비스를 진행하기 위해서 슬라이딩 기법을 지원하는 상황정보 분석 시스템을 제안하였다. 본 논문에서 제안한 시스템은 실시간으로 연속적으로 주위 상황정보를 모니터링하고 일정한 상황조건에서 특정한 조치를 취하는 빌딩의 상태 모니터링 등 많은 USN 응용분야에 적용이 가능하다.

참고문헌

- [1] M. Weiser, "The computer for the Twenty-First Century", Scientific American, pp.94-100, 1991
- [2] B. Schilit, N. Adams, R. Want "Context-Aware Computing Applications", In Proc. Of the 1st International Workshop on Mobile Computing Systems and Applications, December, 1994
- [3] A. Dey, M. Futakawa, D. Salber, G. Abowd, "The Conference Assistant: Combining Context-Aware with Wearable Computing", In Proc. Of the 3rd International Symposium on wearable Computers, IEEE Computer Society Press, San Francisco, pp.21-28, October 20-21, 1999
- [4] D. Apietti, E. Baralis, G. Bruno, T. Cerquitelli. "SAPhyRA: Stream Analysis for Physiological Risk Assessment", IEEE International Symposium on Computer-Based Medical Systems, Maribor, Slovenia, pp.193-198, June 2007
- [5] S. Chandrasekaran, A. deshande, M. Franklin, J. Hellerstein, W. Hong, S. Krishnamurthy, S. Madden, V.

Raman, F. Reiss, M. Shah. "TelegraphCQ: Continuous Dataflow Processing for an Uncertain World", In Proc. of CIDR, January, 2003

[6] <http://berkeley.intel-research.net/labdata/>