Journal of The Korea Society of Computer and Information
Vol. 18, No. 9, September 2013
www.ksci.re.kr
http://dx.doi.org/10.9708/jksci.2013.18.9.111

모바일 상황인식 컴퓨팅에서의 불확실성 관리 기법

김 훈 규*, 원 유 헌*

Uncertainty Management Technology in Mobile Context-Awareness Computing

Hoon-Kyu Kim*, Yoo-Hun Won**

요 약

상황인식 컴퓨팅에서의 불확실성은 주로 상황 획득 메커니즘과 상황처리의 복잡성에 영향을 준다. 상황인식 어플리케이션에 불확실성이 존재하면 어플리케이션에 사용자 만족도에 손상을 주고 이를 쓸모없게 만든다. 본 논문은 상황에 대한 불확실성의 원천을 식별하고, 불확실한 상황정보를 표현하며 이들을 처리하는 방법을 결정할 수 있는 세 가지 전략을 제시하였다. 센서네트워크 시스템은 특성상 시스템 전 과정에 사람의 개입이 없기 때문에 상황정보에 대한 불확실성을 제거하는 노력의 수준에 따라 시스템의 신뢰도에 영향을 주게 된다. 본 논문에서는 제안한 기법을 센서네트워크 시스템 개발에 적용하여, 불확실성 관리가 시스템 개발 수명주기의 한 부분으로 적용이 될 수 있으며, 시스템을 실험한 결과 안정정인 탐지 성능을 나타냄을 확인할 수 있었다.

▸ Keywords : 상황인식 상황 불확실성, 모바일 상황인식 컴퓨팅, 센서네트워크

Abstract

Uncertainty in Context-aware computing is mainly a consequence of the complexity of context acquisition mechanisms and context processing. The presence of uncertainty may harm the users' confidence in the application, rendering it useless. This paper describes a three-phase strategy to manage uncertainty by identifying its possible sources, representing uncertain information, and determining how to proceed, once uncertain context is detected. The level of effort that is necessary to eliminate the uncertainty of context information affects the reliability of the system, because Sensor network system have no intervention of humans. In this paper, We applied proposed method to the development for the sensor network system, Uncertainty management can be applied a part of the system development life-cycle. It confirmed that result of testing show that

•제1저자 : 김훈규
•투고일 : 2013. 8. 2, 심사일 : 2013. 8. 31, 게재확정일 : 2013. 9. 8.
* 홍익대학교 컴퓨터공학과(Dept. of Computer Engineering, Hongik University), 국방과학연구소(ADD)

detection performance is stable.

▸ Keywords : Context-Awareness, Context Uncertainty, Mobile Context-Aware Computin, Sensor Network

I. 서 론

정적이고 잘 정의된 전통적 컴퓨팅 환경과 비교하여 유비쿼터스 컴퓨팅 환경은 동적이며 개방적인 특성을 가지고 있다. 요즘 많이 회자되고 있는 스마트 공간은 지능형 홈, 스마트 회의실, 스마트 자동차와 같이 유비쿼터스 컴퓨팅 기술이 구현된 장소를 의미한다.

스마트 공간에서 사용자는 자신을 둘러싸고 있는 환경 속에 존재하는 많은 장치들과 이들 장치들이 제공하는 다양한 서비스들과 상호작용하기 위하여 다양한 노력을 기울이게 된다. 스마트 공간 안에서 사용자들과 장치들 간의 상호작용을 위해서는 각 장치들과 이들이 제공하는 서비스들이 적절하게 구성되어야 하며, 이들을 제어하고 관리하는 일의 양은 최소화가 되어야 한다. 또한 스마트 공간은 일정 수준 이상의 지능(intelligence)을 가지고 있어야 하며, 제공하는 서비스들은 사용자의 명시적인 입력뿐 아니라 묵시적인 정보들을 이용하여 빠르게 변화하는 사용자의 행동 및 주변 상황을 반영한 서비스를 제공하는 능력이 필요하다. 상황인식 컴퓨팅은 스마트 공간에서 사용자 및 주변의 상황을 인식하여 적정한 서비스를 제공하는 컴퓨팅 패러다임이다.

상황인식에서 상황(Context)은 추상화의 상위수준과 하위수준의 두 가지 차원에서 특징지을 수 있다. 하위수준 상황은 센서로 부터 직접 추론된다. 센서는 센싱하는 하드웨어를 포함한 상황정보를 제공하는 모든 데이터 소스를 의미한다. 예를 들면 온도, 광도, 음향, 움직임, 기압, 환경습도, 사용자 프로파일, 선호도, 스케줄 등과 같이 있다. 반면에 상위수준 상황은 하위수준의 상황으로 부터 추론되는 것으로 상황, 행위, 위치 또는 사용자의 기분 등과 같이 좀 더 추상적이다. 어플리케이션이 신뢰할 수 없는 상황정보를 사용하거나, 상황정보가 애매하게 정의되었거나, 사용된 장비가 부정확하고 불명확한 경우에는 센서의 속성과 상황정보를 식별하고 판단하는데 복잡하고 다양한 기술이 적용된다. 이 경우 상황을 획득하는 과정은 어렵고 복잡한 작업이 된다. 또한 상황식별 과정에서 발생하는 정보의 손실과 유효하지 않은 정보는 상황 생성을 복잡하게 만든다. 이러한 문제는 상황인식 시스템 개발에 중요한 도전 분야로 상황 정보의 불확실한 속성을 다루어야 할 필요성이 제기되는 분야이다[1][2].

상황인식 컴퓨팅에서 불확실성은 불확실한(Uncertainty), 애매한(Ambiguous), 또는 부정확한(Incorrect) 상황 정보로 부터 나타날 수 있다. 불확실성은 어플리케이션이 기능적으로 문제를 일으켜 사용자에게 도움이 되지 못하거나 또는 어플리케이션이 견고하지 못함을 느끼는 원인이 될 수 있다. 어플리케이션에 대한 만족감은 내적 감정이기 때문에 어플리케이션을 사용하는 동안 느끼는 사용자의 긍정적인 경험으로 만족감이 강화되고 부정적인 경험으로 약화된다[3], 따라서 어플리케이션이 좀 더 견고하며 고장을 감내하고 신뢰성을 높이기 위해서는 개발 과정에 불확실성 관리 기법을 포함하는 것이 필요하다. 상황 정보에 존재하는 불확실성 다루기 위하여 주로 베이시안 네트워크나 온톨로지 같은 몇 가지 기법이 제안되었다[4][5][6]. 추론 규칙[7], 퍼지 로직[5][8] 등을 기반으로 하는 다른 기법들도 제안되었으나 이들을 구현하는 데는 많은 작업이 필요하고 때로는 어플리케이션 도메인의 전문가가 되어야 한다.

본 논문은 상황인식 컴퓨팅 환경에서 운용되는 어플리케이션에서 상황의 불확실성 관리에 대한 문제를 기술한다. 2장에서는 불확실성 관리에 대한 기존의 연구를 살펴보고 3장에서는 불확실성에 대한 주요 원인을 기술하고, 불확실성을 분류하며 이를 관리하기 위한 직관적이고 단순한 프로시저를 제안하고 4장에서는 불확실성 관리절차를 실시간 상황감시를 위한 센서네트워크 시스템 개발과정에 적용하여 설명한다. 5장에서 운영실험을 통해 시스템의 성능을 확인하고 6장에서 결론 및 향후 연구내용을 기술한다.

II. 관련 연구

상황인식 컴퓨팅 환경에서 상황의 불확실성을 다루기 위한 몇 가지 접근방법이 제안되어 왔다. Buchholz et al. [8]는 상황의 품질에 대한 개념을 제안했고 이후에 Sheikn et al. [9].에 의해 개념이 확장되었다. 이들은 상황정보가 실세계

현상을 표현하는 정도를 정밀도(precision)로 정의하였고, 어떤 상황이 결정되는 특정 시점에, 상황 인스턴스에 대응되는 실세계 현상을 정확하게 표현하는 확률을 가지고 정확성을 정의하였다. Kim and Lee [10]은 정확성과 완전성이라는 두 가지 단위를 기반으로 상황의 품질을 측정하는 방법을 제안하였다. 이러한 상황요소들은 상황인식 어플리케이션에 의해 사용될 수 있다.

불확실성을 다루는 또 다른 계량기법은 Bayesian networks 이다. TauGu et al. [4]는 Bayesian networks을 사용하여 상황의 불확실성을 처리하는 사용자의 행위를 측정하였고, 확율 값과 관계 링크를 통하여 상황을 정의하기 위하여 온톨로지를 사용 하였다. Truong et al. [6]도 베이지안 네트워크와 온톨로지를 사용하였지만 이들은 상황 온톨리지 정의를 재사용하는데 초점을 두었다. 그러나 각 어플리케이션 도메인에 적합한 온톨로지를 생성하기 위해서는 전문가의 도움이 필요할 뿐만 아니라 어플리케이션 개발자들이 해야 할 일의 양이 엄청나게 많이 늘어난다. 또한 온톨로지를 베이시안 네트워크로의 전환하는 일은 자동화되지 않아서 매우 힘든 작업이 될 수 있다.

Ranganathan et al. [6]는 불확실성을 해결하기 위해 확률적 추론(probabilistic reasoning)과 퍼지 로직을 제안하였다. 그들은 상황을 Context-type(〈Subject〉, 〈Object〉) 와 같은 형식의 술어로 표현한다. 예를 들면 [location(jeff, in, room 3105), activity(room 3102, meeting)] 과 같이 표현할 수 있다. 그러나 이러한 제안은 불확실성 처리를 위한 계량기법에 기초한 것으로 많은 부가적인 작업과 어플리케이션 구현을 위한 정보수집이 필요하다.

이러한 연구들 외에 불확실한 상황이 존재할 때 어플리케이션이 어떻게 동작해야 하는가를 설명하는 연구는 거의 없고 오히려 불확실한 상황을 식별하는 과정에 초점을 맞춘다. 그 외 다른 연구들은 상황 충돌 현상을 정의하고 평가하여 상황정보의 일부를 제거하거나 선택하여 불확실성을 해결하는 솔루션을 사용한다. Bu et al. [11]는 원시 상황(raw context)의 불일치성을 해결하는 알고리즘을 기술하였는데, 불일치성을 해결하는 첫 단계는 충돌을 탐지하는 것으로, 만일 2개의 원시상황 "Tom,walkIn, Room311" and "Tom,walkIn, Aisle3", 가 있다면 Tom이 동시에 두 장소에 있을 수 없으므로 충돌이 탐지될 것이다. 충돌이 일어나면 각 raw context에 대한 상대적인 빈도 값을 계산하고 더 작은 빈도 값을 갖는 원시 상황을 삭제한다. 또 다른 일련의 연구들은 불확실성을 관리하기 위해 사용자가 상황 처리 과정에 직접적으로 관여하도록 한다.

III. 모바일 상황인식 컴퓨팅에서의 불확실성 관리

어떤 일에 대한 명확한 지식이 없을 때, 오류에 대한 걱정 또는 상태가 명확하지 않을 때 불확실성이 발생한다. 일부 연구자들은 불안전 정보를 불확실성에 대한 동의어로 정의하기도 한다. 불완전 정보란 어떤 일이 정의되지 못한 막연함, 상황을 구체적으로 표현하지 못하는 불명확함, 정보의 부족과 관련된 불완전성, 두 가지 대안을 구별할 수 없을 때 존재하는 애매함, 등을 포함한다.

1. 상황인식 컴퓨팅에서의 불확실성

일반적으로 상황에 대한 불확실성은 불확실한 상황(Uncertain context), 애매한 상황(Ambiguous context) 그리고 잘못된 상황(Wrong text)의 세 가지로 개념으로 구분할 수 있다.

1) 불확실한 상황(Uncertain context)

(1) 상황정보를 획득하기 위해 사용되는 장치나 기법을 제대로 통제하지 못하거나 잘 이해하지 못하여 신뢰할 수 없는 소스로 부터 정보가 생성될 때 발생한다.

(2) 어플리케이션의 오작동으로 정보의 유효성이 상실되거나 또는 제공되는 정보의 품질을 사용자가 의심하는 현상에서 발생된다.

불확실성에 대한 이러한 개념은 일반적으로 오류에 의해 야기된다. 예를 들면, Benford et al. [13]는 GPS와 WiFi의 오류로 인해 게임에 참가한 사람의 위치가 특정영역으로 나타났다 사라지고 다시 나타나는 현상으로 인해 시스템에서 제공하는 정보를 사용자가 의심하는 상황을 불확실한 상황으로 표현한다.

2) 애매한 상황(ambiguous context)

(1) 자연어를 사용하여 상황을 정의하는 경우에 발생할 수 있다. Beeharee and Steed[14]는 자연어는 정보의 표현이 때로 매우 추상적이고 실세계와 연계가 어려워 사용자의 이해가 어렵다고 언급하였다. Hightower et al. [15]는 지리적 상황을 사용자가 이해할 수 있는 단어나 개념으로 표현하는 것처럼 상황 정보를 표현하기 어려운 것을 말한다.

(2) 상황 정보가 서로 다른 소스로 부터 얻어질 때 발생할 수 있다. 상황을 획득하는데 서로 다른 기술들을

사용하는 것은 상황의 획득, 처리에 대한 복잡성을 증가시키는 한편 제공된 정보들 간의 모순을 야기 시킨다. Mantyjarvi and Seppanen [16]은 상황 획득의 복잡성으로 인해 서로 다르거나 상충되는 상황들이 발생될 수 있기 때문에 신호 처리 과정의 복잡성을 강조한다. Sheik et al. [9]은 상황을 획득하기 위해 여러 가지 기술을 사용하는 것은 상황을 획득 / 처리과정의 복잡성을 증가시킨다고 지적한다.

(3) 모순, 규칙 위반 또는 불일치가 애매한 상황을 발생시킨다. 예를 들면, 사용자가 동시에 서로 다른 행위를 수행할 때[8], 동시에 동일한 object에 대해 두개 이상의 서비스가 상충되는 오퍼레이션을 수행할 때 [13], 서로 다른 규칙을 활성화 하는 상황이 발생되고 이들 각각이 또 서로 다른 활동을 발생하게 할 때 애매한 상황이 발생된다.

3) 잘못된 상황(Wrong context)

(1) 실험기구, 장치 또는 기술의 정확성과 정밀도가 상황을 획득하는데 중요한 요건이 된다. 예를 들면, GPS가 위치 추정 기술로 가장 널리 쓰이는 기술이지만 여러 가지 요소들로 인하여 부정확할 수 있다[13, 17]. 위치 추정 알고리즘은 무선 네트워크의 신호세기의 강도를 사용하는데 이 기술은 훨씬 더 부정확한 경향이 있다.

(2) 사용되는 장치에 결점이 있거나 데이터가 부정확해서 상황을 추론하는데 충분한 정보가 없는 현상과 같이 정보의 부족이나 결핍은 불확실성의 또 다른 요인이 된다. 위치정보가 부정확하여 어플리케이션의 동작이 일정하지 않을 때 이러한 부정확한 정보를 잘못된 상황이라고 한다.

(3) 실효된 정보 : 특정한 시점에서 사용되어야 하나 그 시점에 유효하지 않은 정보를 말하며, Sheikh et al. [9]는 상황의 최신화는 적정한 품질의 상황획득을 위해 고려해야 할 중요한 요소라고 언급하였다.

2. 불확실성(Uncertainty) 분류

상황인식 어플리케이션을 개발할 때 모듈화, 확장성, 유연성과 재사용을 위해 상황 획득 기능과 획득한 상황을 처리하는 기능을 명확히 분리할 수 있는 아키텍처를 사용하는 것이 필요하다. 예를 들면 Baldauf et al. [18]는 계층기반 상황인식 어플리케이션 구조를 제안하였다. 여기서 제안한 계층은 센서, 원천 데이타 조회, 전처리, 상황 정보 관리, 상황인식 어플리케이션 등으로 구성된다. 하부의 세 계층은 상황 획득 과정에 관련되고, 상위 두 계층은 상황정보의 사용과 관계된다. Baldauf et al. [18]가 제시한 계층구조를 기반으로 생각해 보면 상황 어플리케이션에서의 불확실성은 두 가지 타입으로 정의할 수 있다.

1) 원천적 불확실성(Original Uncertainty) : 상황에서의 근본적인 불확실한 상황을 의미하는 것으로 원시 상황을 획득하고 생성하는 기법이나 기술과 밀접하게 연관이 된다. 이러한 불확실성은 어플리케이션이 센서로부터 자신의 상황을 설정할 때 나타난다. Baldauf et al 가 제안한 구조에 따르면 데이터를 획득한 후 상황처리를 즉시 할 수 있게 하고, 이후에 어플리케이션에서 사용할 수 있도록 준비하는 전처리 계층은 원천적 불확실성 관리 기법이 반드시 포함되어야 하는 계층이다.

2) 파생된 불확실성(Drived Uncertainty) : 파생된 불확실성이란 상황 정보 처리 결과에 발생되는 불확실성을 의미하는 것으로, 상황을 사용하는 부분과 밀접하게 관련되어 있다. 이러한 불확실성은 어플리케이션이 새로운 상황에 대한 행위를 수행할 때 나타나는 것으로 어플리케이션의 상황인식 능력으로도 볼 수 있다. 이러한 관점과 Baldauf et a 에 의해 제안된 구조에 따르면, 어플리케이션 계층은 획득된 상황에 대응되는 서비스로 정의하기 때문에 어플리케이션 계층에는 이러한 유형의 불확실성을 다루는 기법이 포함되어야 한다. 불확실성에 대한 이러한 분류는 불확실성이 발생될 때, 좀 더 단순하고 구체적인 솔루션을 통해 상황을 식별하고 처리하는데 도움이 된다.

3. 불확실한 상황의 영향

상황의 불확실성은 상황인식 어플리케이션에 다양한 방법으로 영향을 미칠 수 있다. 비록 이러한 영향이 어플리케이션의 도메인에 따라 달라지기는 하지만 불확실하고 애매하며 잘못된 상황정보들은 어플리케이션에 심각한 영향을 줄 수 있는 여러 현상들이 존재한다. 예를 들면, Broens et al. [19]은 사용자의 심박, 움직임과 위치를 측정하는 인체센서를 사용하는 간질 환자를 위한 상황인식 어플리케이션인 원격 홈 케어 시나리오를 설명한다. 저자는 상황정보가 갖는 문제가 시스템을 무용지물로 만들 수 있고, 사용자에게 심각한 상해를 야기시킬 수도 있으며 심지어는 죽게 할 수 도 있다고 주장한다. 그러나 재고관리시스템 등과 같은 어플리케이션은 불확실성에 의한 영향의 심각성이 크지 않은 경우도 있다.

4. 불확실한 정보의 관리

상황인식 컴퓨팅에서 시스템의 신뢰성 확보를 위해서는 어플리케이션을 설계하는 과정에 상황에 대한 불확실성을 관리하는 절차가 반드시 고려되어야 한다. 본 논문에서는 어플리케이션 설계자가 상황의 불확실성으로 인해 발생할 수 있는 위험을 이해하고 이를 식별하는데 도움을 주며, 어플리케이션에서의 상황의 불확실성을 통제하기 위한 체계적인 기법을 제안한다.

본 논문에서 제안하는 상황의 불확실성을 관리하기 위한 기법은 불확실성의 식별(Identification), 측정(Measurement), 처리(Treatment)의 3단계로 구성된다.

1) 불확실성 식별(Uncertainty Identification)

어플리케이션이 원천적 불확실성 또는 파생된 불확실성을 생성하는지 여부를 확인하는 것으로 상황 추정 과정에 대한 상세한 분석이 필요하다. 불확실성에 대한 잠재적인 원천을 찾기 위하여 상황이 획득되고 처리되는 과정을 연구하고 어플리케이션의 운영상의 제약점에 대해 알아낸다.

원천적 또는 파생된 불확실성을 식별하기 위해 확인해야할 점검목록을 정리하면 다음과 같다.

- ㅇ 원천적 불확실성 식별을 위한 점검 목록
  - 어플리케이션에서 사용될 상황 요소의 유형 : 상황의 유형은 일반적으로 물리적 상황(빛, 소음, 교통조건, 기온 등), 시간 상황(일, 주, 월 등 시간 정보), 사용자 상황(사용자 프로파일, 위치, 주변사람 등) 및 컴퓨팅 상황(네트워크 연결성, 통신 비용, 통신 대역폭, 워크스테이션 등과 같은 자원들) 등이 있다.
  - 각각의 상황 요소를 위해 사용될 센서의 유형 : 온도 센서, 자기센서, 진동센서 등의 센서 유형이 있다.
- ㅇ 상황의 원천에서 발생할 수 있는 잠재적 불확실성 위험을 알기 위한 목적으로 사용되는 각 센서들에 대한 점검 목록
  - 센서에 사용되는 기술
  - 센서 내부의 오퍼레이션
  - 센서의 정확도와 정밀도
  - 센서의 성능
- ㅇ 파생된 불확실성에 대한 정보를 수집하기 위한 점검항목은 어플리케이션이 상황으로 무엇을 어떻게 하는가에 대한 것을 이해하고 식별하기 위한 것이다. 추론 프로세스, 규칙의 사용, 정보의 변경, 인터페이스 적용 또는 상황의 사용을 포함하는 다른 프로시저 등이 고려되어야 한다.
  - 어플리케이션에서 상황이 사용되는 방법 : 다른 상황과 결합하여 새로운 상황을 발생하는지, 단일 상황으로 서비스를 제공하는지 등
  - 상황의 변화에 따라 사용자 인터페이스와 어플리케이션 행위에서 변경되는 내용 : 시간의 변화에 따라 다른 사용자 인터페이스를 디스플레이하거나 제공되는 서비스의 변경내용 등

2) 불확실성 판단(Uncertainty Measurement)

불확실성을 표현하는 방법을 제안하고 불확실성을 명확하게 하기 위한 기법을 사용한다. 이전 단계에서 식별된 불확실성을 생성하는 각 요소들을 다루기 위한 규칙을 생성한다.

3) 불확실성 처리(Uncertainty Treatment)

불확실성을 실제로 처리하기 위한 행위를 만든다. 이들 행위들은 이전 단계에서 정의된 규칙들 중 하나가 트리거 될 때 실행된다. 실행될 행위들은 두 그룹으로 분류된다.

- 불확실성에 대한 자동처리(Automatic handling of uncertainty) : 어플리케이션이 사용자의 간섭 없이 수행되는 경우에 불확실성이 자동으로 처리된다. 어플리케이션에서는 불확실성에 대한 재평가와 부가적인 정보 소스를 사용할 수 있다. 재평가 과정에는 추가의 시간과 노력이 필요하며, 부가적인 정보소스는 어플리케이션의 복잡성을 증가시키지 않고 직접적으로 얻어질 수 있다.
- 사용자 지원에 의한 불확실성 처리(User-assisted uncertainty handling) : 사용자가 불확실성의 존재를 인식하여 불확실성을 처리하는 과정에 관여하게 된다. 이 경우 사용자에게 불확실성의 존재를 알리고 부가적인 정보 또는 사실 확인을 요청하거나, 사용자에게 불확실성의 존재를 알리고 사용자가 직접 적절한 행위를 결정을 허용하게 한다.

Ⅳ. 센서네트워크 시스템에서의 불확실성 관리

유비쿼터스 응용분야 중 하나로 유비쿼터스 기술의 핵심 인프라인 센서네트워크 기술은 상황인식 컴퓨팅의 좋은 범주에 속한다. 센서네트워크는 다수의 센서노드들로 구성된다. 센서노드들은 센서 모듈을 통해 상황 정보를 획득, 처리하여 무선 통신을 이용하여 이웃 노드나 기지국으로 상황 정보를 전송하게 된다. 센서네트워크는 접근이 어려운 감시 취약지역

이나 특정 감시 지역에 대한 무인감시 기능을 제공하여 감시 경계의 새로운 영역으로 인식된다. 군사응용 분야에서는 적 부대의 위치, 구성, 이동방향 등에 대한 근접 탐지 및 추적 임무, 무인으로 원격 센싱 정보수집 및 모니터링을 통한 감시 정찰, 최전방초소 철책선 침입탐지, 아군 주둔지 및 주요 시설경계, 생화학, 방사능, 핵 오염지역 탐지, 아군 병사 및 주요 장비의 위치와 상태 파악 등에 활용이 가능하다. 민간 분야에서는 화훼단지, 특용작물 지대 등에 설치되어 동물 또는 외부인의 침입을 감시하는 용도로 활용될 수 있다.

본 장에서는 상황인식 컴퓨팅 분야의 주요한 범주에 속하는 실시간 상황감시를 위한 센서네트워크 시스템 개발과정에 본 논문에서 제안한 불확실성 관리기법을 적용하여 불확실성을 식별하고 제거함으로 센서네트워크 시스템의 신뢰성을 향상시키고자 한다.

1. 어플리케이션의 분석 및 설계

센서네트워크 시스템의 센서 노드는 환경 센서를 비롯한 PIR, 음향, 진동, 자기 센서 등 복합 센서를 탑재하고 있으며, 센서 신호의 처리 및 융합을 통해 침입 상황정보를 생성한다.

센서네트워크 시스템에서 침입 탐지 상황을 결정하는 과정은 크게 두 단계로 구분할 수 있다. 첫 번째 단계는 센서 노드 레벨에서의 신호처리이고 두 번째 단계는 센서필드 레벨에서의 상황 정보 융합이다.

센서네트워크 시스템에서 운용되는 센서노드는 컴퓨팅 자원이 작고, 저 전력으로 운용이 되기 때문에 신호처리 과정에 필요한 많은 일들을 수행할 수 없어서 센서노드 단과 센서필드 단으로 기능을 분산할 필요가 있다. 또한 노드 사용시간을 늘리고 오탐지율을 낮추고 탐지율을 향상하기 위해 기능 분산 및 상황정보 융합 등의 다양한 방법들이 요구된다. 그림 1은 센서네트워크 시스템의 운영 개념도이다.

센서노드는 일정시간 간격으로 센서를 통해 노드주변의 상황 값을 획득한다. 획득된 센싱 값은 변환 및 필터링 과정을 거쳐 잡음과 오류 등을 제거한 후 침입탐지 상황 여부를 결정하는 일련의 과정이 수행된다. 이후에 연속적으로 입력되는 다양한 상황 정보들을 융합하여 침입여부를 결정하게 되고 확정된 침입으로 확정된 상황 정보는 서버로 전달하게 된다.

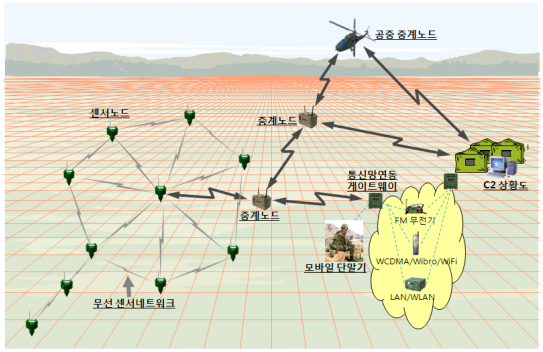

그림 1. 센서네트워크 운영개념도
Fig. 1. Sensor Network system Configuration

서버에서는 각 센서노드로 부터 전달받은 침입탐지 상황정보를 기반으로 센서필드 수준에서 침입을 판단하고 최종적으로 침입 상황을 결정하여 상황 정보를 사용자 단말기에 디스플레이 하며 이후 지속적으로 입력되는 침입 탐지 상황을 관리하게 된다. 표 1은 센서노드가 수집하는 상황정보의 세부 속성을 나타낸다.

표 1. 센서노드 상황정보 속성
Table 1. Context Information Property

구 분	속 성
상태 정보	• 노드위치 : 경/위도 좌표 • 부모노드 : 노드 ID • 센서 종류 : PIR/진동/음향/자기 • 센서전원 상태 : on/off • 노드 배터리 상태 : 잔류량 %
탐지 정보	• 탐지노드 : 노드 ID • 탐지시간 : 년/월/일/시/분/초/밀리미터초 • 탐지표적 : 인원/미확인 • 탐지센서 : 에너지 값 • 탐지센서 : 임계값

센서네트워크 시스템은 침입자를 탐지, 식별하는 과정부터 사용자의 단말기에 탐지 결과를 전시하는 전체과정에 사용자가 개입하지 않으며 사용자는 전시되는 내용으로 감시지역에 대한 실시간 상황을 판단하게 된다. 센서네트워크 시스템의 신뢰성이 높으면 탐지된 표적에 대한 자동 타격 시스템 까지도 연계될 수 있겠지만 상황에 대한 불확실성 등으로 인해 센서네트워크 시스템의 활용성에는 한계가 있다.

센서네트워크 시스템에서는 센서로 부터 입력되는 센싱 값 자체가 불확실한 상황이 될 수 있다. 실제로 센서노드가 탐지한 센싱 값만으로는 침입 탐지여부에 대한 확신이 없기 때문이다.

센서노드에서 일련의 과정을 거쳐 불확실한 상황이 어느정

조 정제된 상황으로 생성 되지만, 생성된 상황정보 또한 불확실성이 내포되어 있을 수 있기 때문에, 서버에서는 상황 정보 융합 과정을 거쳐 최종적인 상황 정보를 생성한다.

이러한 불확실한 상황들에 대해 적절한 관리 및 처리를 통해 신뢰할 수 있는 상황을 생성하면 시스템은 안정적인 서비스를 제공하며 시스템의 성능과 신뢰성을 향상시킬 수 있다.

2. 불확실성의 처리

비가시 지역 또는 무인 지역에 대한 상황 감시를 통해 외부 침입자 또는 환경의 변화를 판단하는 센서네트워크 시스템의 속성상 시스템 운영의 전 과정에 불확실성이 내포되어 있다.

센서 네트워크는 노드 결함, 무선통신 장애, 노드 위치 변경, 환경 변화 등 내외부적인 영향에 취약하며, 센서 데이터는 쉽게 결함(또는 오류)을 포함할 수 있다. PIR 센서의 경우 햇빛, 바람, 기온, 습도의 영향을 받으며, 회로판에서 발생하는 전자기 잡음은 SNR(signal noise ratio)을 낮춰 신호와 잡음을 구별하는 센싱 알고리즘에 문제를 발생시킨다. 또한 주변 온도의 변화에 따라 센서 신호도 함께 변하는 열 이동(thermal drift) 현상이 발생하고, 무선 전송이 자기계 센싱 회로에 간섭을 일으켜 데이터 오류를 발생시킨다. 이러한 다양한 문제점들이 센서네트워크 시스템의 불확실성으로 간주할 수 있다.

센서네트워크 시스템에서의 불확실성 처리는 침입탐지 상황을 결정하는 단계와 같이 센서노드 레벨과 센서필드 레벨의 두 단계로 구분된다. 센서노드 레벨에서는 노드 내에 존재하는 원천적 불확실성과 상황정보를 생성하면서 나타나는 파생된 불확실성을 처리하여 상황정보의 신뢰성을 높일 수 있다. 센서필드 레벨에서는 노드에서 생성된 상황 정보와 이웃 노드들의 상황정보들의 융합을 통해 파생된 불확실성을 처리하므로서 정확한 상황정보를 생성하고, 그에 맞는 서비스를 사용자에게 제공할 수 있다. 3.4절에서 정의한 불확실성 관리기법을 센서네트워크 시스템 개발에 적용하면 다음과 같다.

1) 불확실성 식별

- 사용될 센서 유형 식별 : 센서노드에서 사용되는 센서는 오류가 발생할 수 있기 때문에 적용될 센서들은 실험과 분석을 통하여 성능이 우수하고 안정적인 센서를 선정한다. 센서네트워크 시스템에서는 진동, 음향, 자기, PIR 등을 사용한다.
- 어플리케이션에서 사용될 상황요소의 유형 : 센서로부터 탐지되는 물리적 상황, 시간 상황, 컴퓨팅 상황 등의 상황 유형을 사용한다.

2) 불확실성 판단

센서노드 레벨에서는 센서에 대한 침입 상황은 센싱된 정보와 임계값을 비교하여 센싱값이 침입상황인지를 판단한다. 각 센서들마다 탐지 거리 능력이 다르기 때문에 시간의 연속성과 센서의 탐지거리 기준으로 침입자의 이동 방향과 패턴에 따른 규칙을 기반으로 센서별 탐지 상황을 융합하여 불확실성을 제거, 센서노드 레벨의 침입탐지 상황을 결정한다.

센서필드에서는 이웃 센서 노드들의 탐지 결과를 융합하여 탐지 상황 정보의 신뢰성을 향상한다. 침입자가 센서필드 범주내로 들어와서 이동을 하게 되면 이동경로 상에 배치된 센서노드들의 탐지 정보를 일정한 규칙을 기반으로 탐지 여부를 판단하게 되는데 이 규칙은 간단한 모델링 또는 다양한 시나리오를 기반으로 휴리스틱하게 정의할 수 있다.

3) 불확실성 처리

센서네트워크 시스템의 특성상 불확실성 처리 전체 단계에 사람의 관여 없이 자동으로 처리가 된다. 1단계에서 식별된 상황 요소들에 대하여 2단계에서 정의된 여러 규칙들이 일련의 처리 절차에 적용되어 다양한 불확실성이 제거되고 시스템의 정확한 결과를 산출한다.

상황인식 어플리케이션 관점에서 보면 불확실성 처리 또한 센서노드 레벨과 센서필드 레벨로 구분하여 처리된다. 센서노드에서의 불확실성 처리 절차는 그림2와 같다.

그림 2 센서필드에서의 불확실성 처리 절차
Fig. 2. Procedure for Uncertainty processing in sensor node

그림 3은 센싱 값에서 특징점을 추출하고 표적을 탐지, 분류하는 절차를 의사코드로 기술하였다. 특징점 추출을 위해서는 pitch-based ration 알고리즘을 적용하였고 FFT를 통해 표적 탐지에 필요한 정보를 산출하였다.

```
GetSeizmic(); // * 진동센서 표적 탐지  *//
currBuffer; currSeizmic; fftSeizmic; detTrue;
currFeature;
InitDetection();//* 센싱값 획득을 위한 초기화 *//
currSeizmic = GetSeizmic(); //* 진동센서로부터 센싱 값 획득
*//
currFeature = GetFeture(currSeizmic);//*특징점 추출*//
if currFeature 〉 V_feature
  currBuffer = fftSeizmic(currSeimic);
  if detPeople(currBuffer)
    detTrue = true;
 //* 특징점 추출 결과가 임계범위에 들어오면  FFT를 통해 침입자를
분류하여 탐지 결과 결정*//
```

그림 3. 진동센서에 의한 표적 탐지 프로시져
Fig. 3. Target Detection Procedure Using Seizmic Sensor

위의 처리과정을 통해 침입 상황에 대한 불확실한 요소들이 제거되고 센서노드는 침입자에 대한 탐지 결과를 탐지 상황으로 생성하게 된다.

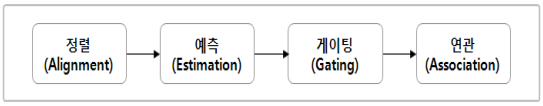

그림 4 센서필드에서의 불확실성 처리 절차
Fig. 4. Procedure for Uncertainty processing in sensor field

센서노드에서 식별된 탐지 상황에서 불확실성을 제거하여 좀 더 정확한 침입탐지 상황을 생성하기 위하여 센서필드 상에서의 상황정보 융합 과정을 통해 침입탐지 상황을 결정하게 된다. 그림4는 센서필드 상에서의 불확실성을 처리하기 위한 정보융합 절차를 나타낸 것이다.

표3 은 각 단계의 주요 역할 및 적용하는 알고리즘을 설명하였다.

표 3. 단계별 역할 및 적용 알고리즘
Table 3. Action and Algorithm

구분	역할	알고리즘
정렬	탐지정보를 단위, 공간, 시간 및 차원으로 정렬	
예측	정렬된 표적에 대해서 최적예상 위치 산출	칼만 필터
게이팅	표적 예측위치를 기준으로 속성에 따라 최적 예상 크기 및 모양 산출	Circular
연관	표적의 예측 위치와 센서 탐지 위치와의 비교를 통한 최적 연관 수행	Nearest Neighbor

V. 운영 실험

본 장에서는 개발된 센서네트워크 시스템에 대한 운영실험을 통해 성능을 검증한다.

본 실험에는 음향, 자기, 진동 및 PIR 센서가 장착된 12개의 센서노드가 사용되었고, 표 4와 같은 센서노드를 사용하였고 센서필드 구성은 그림 5와 같다.

실험은 사람의 침입을 탐지하는 것을 목적으로 하였으며, 먼저 침입탐지 판단 임계값 설정을 위해 환경 노이즈를 측정하였다. 운영실험에서는 오경보율과 탐지율을 측정한다. 오경보율은 1시간 동안 탐지가 일어나지 않은 상황에서 발생하는 탐지의 횟수로 나타내며 탐지율은 발생된 총 탐지 이벤트에 대한 탐지 성공률을 나타낸다. 탐지 상황 정보에 내재된 불확실성 정보를 잘 통제하지 못하면 오경보율이 높아지고 탐지율은 낮아지게 된다. 본 논문에서는 오경보율 측정을 위해 1회 1시간씩 5회에 걸쳐 실험을 수행하였고, 탐지율 측정을 위해서는 10회 이벤트 발생을 1세트로 하여 총 5세트 실험을 실시하였다.

표 4. 센서노드 H/W 및 S/W 구성
Table 4. H/W, S/W Specification

구 분		구성품	형상
MCU		MSP430	
RF		CC2420	
적용 센서	PIR	AMN24111	
	음향	WP23502	
	진동	GS-20DX	
	자기	HMC1002	
OS		초소형 OS	

실험결과는 표 5 와 같다. 센서네트워크 시스템은 상황정보에 존재하는 불확실성 관리 노력 정도에 따라 시스템의 성능 및 안정성에 영향을 준다. 그러나 안정성 확보를 위해 지나친 오버헤드를 초래할 수 있기 때문에 상황정보의 불확실성 관리에 대한 절대 기준을 제시하기가 어렵고 시스템의 목적에 맞는 절충이 필요하다. 그렇기 때문에 시스템의 환경, 컴퓨팅 자원 등에 대한 검토 없이 유사 시스템과의 성능을 단순 비교하는 것은 의미가 없다.

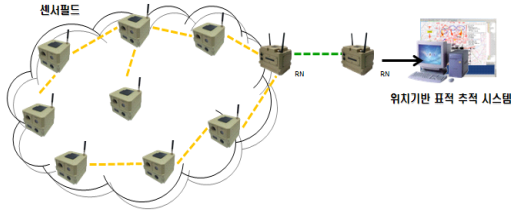

그림 5 실험환경 구성도
Fig. 5. Testing Configuration

표 5. 운영 실험 결과
Table 5. Result for Operation Test

구분	기준	실험결과
오경보율	1시간	1회 이내
탐지율	10회/1세트 이벤트 발생	90%

VI. 결론 및 향후 연구

상황인식 컴퓨팅에서 사용하는 상황정보에 존재하는 불확실성은 시스템의 운용에 영향을 줄 수 있기 때문에 상황인식

어플리케이션 내에서 불확실성을 식별하고 처리하여 시스템의 신뢰성을 향상하는 것이 중요한 과제이다.

본 논문은 상황인식 컴퓨팅에서 다루어야 할 상황의 불확실성에 대하여 정의하고, 불확실성에 대한 이해를 높이기 위하여 불확실성을 분류, 설명하였고 이러한 불확실한 정보로 인하여 상황인식 어플리케이션에서 불확실성이 발생될 수 있음을 기술하였다. 또한 상황인식 어플리케이션을 구현하기 위하여 반드시 관리되어야 할 불확실성을 식별, 판단, 처리하는 메커니즘을 제안하였고 이를 센서네트워크 시스템에 적용하여 상황인식 어플리케이션의 구현의 신뢰성 향상에 기여하였다.

본 논문에서 제안된 불확실성 식별, 판단, 처리 기법을 센서네트워크 시스템의 구현에 적용하였으나 불확실성을 식별하고 처리하는 구체적인 방법은 센서네트워크 시스템이라 할지라도 그 적용 범위와 노드의 성능 등에 따라 매우 다양하여질 수 있다. 중요한 것은 상황인식 컴퓨팅 환경에서 운영되는 상황인식 어플리케이션은 적용 도메인에서 발생 할 수 있는 불확실한 상황정보를 사전에 식별하여 적절히 통제/관리하는 것이다. 센서네트워크 시스템에서 센서 필드 상에서 복합 센서노드의 상황융합을 통한 불확실성 제거, 임계값 자동계산을 통한 지능화된 침입 상황 식별 등과 같은 기법들에 대하여 추가적인 연구가 수행되면 좀 더 안정적이고 수준 높은 센서네트워크 시스템이 될 것으로 생각된다.

참고문헌

〔1〕 Satyanarayanan, M. Pervasive computing: Vision and challenges. IEEE Personal Communications, 8(4), 10-17. 2001.

〔2〕 Satyanarayanan, M. Coping with uncertainty. IEEE Pervasive Computing, 2(3), 2. 2003.

〔3〕 Antifakos, S., Kern, N., Schiele, B., & Schwaninger, A. Towards improving trust in contextaware systems by displaying system confidence. In 7th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCIG05) (pp. 9-14). ACM. 2005.

〔4〕 Gu, T., Pung, H. K., & Zhang, D. Q. A bayesian approach for dealing with uncertain contexts. In 2nd International Conference on Pervasive Computing (Pervasive 2004). 2004.

〔5〕 Ranganathan, A., Al-Muhtadi, J., & Campbell, R. H. Reasoning about uncertain contexts in Pervasive computing environments. IEEE PERVASIVE Computing Journal, 3(2), 62-70. 2004.

〔6〕 Truong, B. A., Lee, Y.-K., & Lee, S.-Y. Modeling and reasoning about uncertainty in context-aware systems. In IEEE International Conference on e-Business Engineering (ICEBE'05) (pp. 102-109). IEEE Computer Society. 2005.

〔7〕 Dey, A. K., Mankoff, J., Abowd, G. D., & Carter, S. Distributed mediation of ambiguous context in aware environments. In Proceedings of the 15th Annual ACM Symposium on User Interface Software and Technology (pp. 121-130). ACM Press. 2002.

〔8〕 Buchholz, T., Kpper, A., & Schiffers, M. Quality of context: What it is and why we need it. In 10th Workshop of the Open View University Association (OVUA 03). 2003.

〔9〕 Sheikh, K., Wegdam, M., & Sinderen, M. V. Quality-of-context and its use for protecting privacy in context aware systems. Journal of Software, 3(3), 83-93. 2008.

〔10〕 Kim, Y., & Lee, K. A quality measurement method of context information in ubiquitous environments. In Hybrid Information Technology (ICHIT 06) (pp. 576-581). IEEE Computer Society. 2006.

〔11〕 Bu, Y., Gu, T., Tao, X., Li, J., Chen, S., & Lu, J. Managing quality of context in pervasive computing. In Sixth International Conference on Quality Software (QSIC 2006). IEEE Computer Society. 2006.

〔12〕 Park, I., Lee, D., & Hyun, S. J. A dynamic context-conflict management scheme for group-aware ubiquitous computing environments. In 29th Annual International Computer Software and Applications Conference (COMPSAC 2005) (pp. 359-364). IEEE. 2005.

〔13〕 Benford, S., Crabtree, A., Flintham, M., Drozd,

A., Anastasi, R., Paxton, M., Tandavanitj, N.,Adams, M., & Row-Farr, J. Can you see me now?. ACM Transactions on Computer-Human Interaction, 13(1), 100-133. 2006.

〔14〕 Beeharee, A., & Steed, A. Exploiting real world knowledge in ubiquitous applications. Personal and Ubiquitous Computing, 11, 429-437. 2007.

〔15〕 Hightower, J., Consolvo, S., LaMarca, A., Smith, I., & Hughes, J. Learning and recognizing the places we go. In Ubiquitous Computing: 7th International Conference on Ubiquitous Computing, UBICOMP 2005 (pp. 159-176). Springer. 2005.

〔16〕 Mantyjarvia, J., & Seppanen, T. Adapting applications in handheld devices using fuzzy context information. Interacting with Computers, 15(4), 521-538. 2003.

〔17〕 Benford, S., Anastasi, R., Flintham, M., Drozd, A., Crabtree, A., Greenhalgh, C., Tandavanitj, N. ,Adams, M., & Row-Farr, J. Coping with uncertainty in a location-based game. IEEE Pervasive Computing, 2(3), 34-41. 2003.

〔18〕 Baldauf, M., Dustdar, S., & Rosenberg, F. A survey on context-aware systems. International Journal of Ad Hoc and Ubiquitous Computing, 2(4), 263-277. 2007.

〔19〕 Broens, T., Halteren, A. V., Sinderen, M. V., & Wac, K. Towards an application framework for context-aware m-health applications. International Journal of Internet Protocol Technology, 2(2), 109-116. 2007.

저 자 소 개

김 훈 규

1986 : 홍익대학교
전자계산학과 이공학사.
1992 : 홍익대학교
전자계산학과 이공학석사.
현 재 : 홍익대학교
컴퓨터공학과 박사과정.
국방과학연구소 책임연구원
관심분야: 소프트웨어 공학,
상황인식 컴퓨팅,
Email : hunk@add.re.kr

원 유 헌

1972 : 성균관대학교
수학과 이학사.
1975 : 한국과학기술원
전자계산학과 이학석사.
1985 : 고려대학교
전자계산학과 이학박사.
현 재 : 홍익대학교
컴퓨터공학과 교수
관심분야: 프로그래밍 언어론, VoIP,
네트워크 보안
Email : yhwon@hongik.ac.kr