

# 유비쿼터스 컴퓨팅 시스템을 위한 상황인식 구조에 관한 연구

두경민, 지삼현, 김순국, 이강환

## A study on the Context-Aware Architecture for Ubiquitous on Computing System

Kyoungmin Doo, Samhyun Chi, Sunguk Kim, Yun Chen, Kangwhan Lee

Korea University of Technology and Education  
dkm0303@kut.ac.kr

### 요 약

Ubiquitous Computing System란, 언제 어디서나 통신 및 컴퓨팅이 가능하고 컴퓨팅 시스템이 상호간에 정보를 공유하고 협력하는 컴퓨팅 시스템이다. 이로써 기존의 컴퓨팅 환경과 같이 사용자와 컴퓨터간의 대화형 상호작용이 아닌 물리적인 환경 상황(Context)등을 시스템이 스스로 인식하고 이를 기반으로 사용자와의 상호 작용을 지원하는 상황인식 기술이 필수적인 요소로 부각되고 있다.

Ubiquitous Computing System을 위해 사용자 및 주변 환경의 정보를 감지하는 센서(Sensor) 기술이 필요하다. 하지만 사용자 및 주변 환경으로부터 입력되는 불확실하거나 모호한 상황정보에 대한 표현과 추론에 대한 연구는 부족한 실정이다.

본 논문은 Rule based System을 활용하여 CRS(Context Recognition Switch)라는 새로운 개념을 도입한 Context Aware Architecture를 제시한다.

CRS는 유비쿼터스 컴퓨팅 시스템을 위해서는 센서로부터 복합적으로 인지된 사용자 정보 및 주변 환경의 정보를 사용자로부터 수동적으로 설정되거나 System의 지속적으로 수집된 정보의 통계 값인 Reference Value와 비교하여, 각 상황에 따른 개별적이고 특화된 서비스를 실행을 하도록 제공한다. 이로써 같은 정보의 입력이 들어와도 그 주변 환경의 상황에 따라 사용자의 필요에 최적화된 실행을 할 수 있다.

마지막으로, Ubiquitous Computing System의 향후 발전 가능성을 예상하고, 본 논문에서 제시한 Context Aware Architecture의 유용성을 짐작해 본다.

Key World: Ubiquitous Computing System, Context Aware Architecture, Context Recognition Switch

### 1. 서 론

최근 컴퓨터 시스템들은 인간 생활의 편리성을 위해 다양한 곳에서 하나의 네트워크에 연결되고 있으며, 더불어 언제 어디서나 통신 및 컴퓨팅이 가능하고 컴퓨팅 시스템이 상호간에 정보를 공유하고 협력하는 유비쿼터스 컴퓨팅으로 발전하고 있다.

이로써 기존의 컴퓨팅 환경과 같이 사용자와 컴퓨터간의 대화형 상호작용이 아닌 물리적인 환경 상황(Context)등을 시스템이 스스로 인식하고 이를 기반으로 사용자와의 상호 작용을 지원하는 상황인식 기술이 필수적인 요소로 부각되고 있다.

유비쿼터스 컴퓨팅 시스템을 위해서는 사용자 및 주변 환경의 정보를 감지하는 센서(Sensor) 기술이 필요하다.

기존의 키보드나 마우스 등의 컴퓨터 인터페이스를 극복하기 위해 표정, 동작, 음성, 생체 인식 등의

사용자 중심 인터페이스 기술이 필요하다.

하지만 사용자 및 주변 환경으로부터 입력되는 불확실하거나 모호한 상황정보에 대한 표현과 추론에 대한 연구는 부족한 실정이다.

본 논문은 이러한 문제를 차세대의 유비쿼터스 시스템에서 해결하기 위해 Rule based System을 활용하여 CRS(Context Recognition Switch)라는 새로운 개념을 도입한 Context Aware Architecture를 제시한다.

Rule based System은 인공지능 분야에서 성공적으로 실제 응용 분야에 적용된 기법으로 그 우수성이 잘 알려져 있다.

하지만 기존의 소프트웨어 알고리즘에 기초한 Rule based System의 처리 속도는 Context Aware Computing System과 같은 실시간 처리가 필요한 응용 분야에는 적합하지 않다.

이를 해결하기 위해 CRS를 도입하여 새로운

System Architecture를 제시한다.

CRS는 유비쿼터스 컴퓨팅 시스템을 위해 센서로부터 복합적으로 인지된 사용자 정보 및 주변 환경의 정보를 이미 설정된 Reference Value와 비교하여, 각 상황에 따른 개별적이고 특화된 서비스를 실행을 하도록 돕는다.

이로써 같은 정보의 입력이 들어와도 그 주변 환경의 상황에 따라 사용자의 필요에 최적화된 실행을 할 수 있다.

본 논문의 구성은 구성은 다음과 같다. 제 2장에서는 센싱 및 인식 기술을 통해 사용자가 처한 환경을 컴퓨터가 인식하는 상황인식 기술의 개념을 정의하고, 3장에서는 Context Aware Computing의 UoC(Ubiquitous on Chip) 개발을 위한 Architecture를 소개한다. 마지막 결론에서는 CRS를 활용한 Architecture가 향우의 Ubiquitous Computing System에 활용될 수 있는 여러 가지 예를 통해, 그 발전 가능성과 그 이상의 유용성을 짐작한다.

## II. 상황 인식 컴퓨팅

Ubiquitous Computing System란 언제 어디서나 통신 및 컴퓨팅이 가능하고 컴퓨팅 시스템이 상호간에 정보를 공유하고 협력하는 컴퓨팅 시스템이다. 이로써 기존의 컴퓨팅 환경과 같이 사용자와 컴퓨터간의 대화형 상호작용이 아닌 물리적인 환경 상황(Context)등을 시스템이며, 스스로 인식하고 이를 기반으로 사용자와의 상호 작용을 지원하는 상황인식 기술이 필수적인 요소로 부각되고 있다.

Context Aware Computing System은 주변 환경을 여러 센서를 통해 감지하여 Context를 파악 및 인식하고, 그 인식된 정보에 의해 최적화된 서비스를 제공하는 시스템이다.

Ubiquitous Computing System을 위해 사용자 및 주변 환경의 정보를 감지하는 센서(Sensor) 기술이 필요하다. 하지만 사용자 및 주변 환경으로부터 입력되는 불확실하거나 모호한 상황정보에 대한 표현과 추론에 대한 연구는 부족한 실정이다.

본 논문은 Rule based System을 활용하여 CRS(Context Recognition Switch)라는 새로운 개념을 도입한 Context Aware Architecture를 제시한다. 또한, CRS의 Reference Value인 평균 값을 Ubiquitous Computing System에 보다 효율적으로 산출해 내는 수식을 설명한다.

## III. 유비쿼터스 컴퓨팅 시스템을 위한 상황 인식 구조 제안

본 논문은, 상황인식 기술을 SoC로 구현하여 Ubiquitous Computing을 실현하고자 새로운 UoC

Architecture를 제안한다. UoC를 위해 제안된 Architecture는 다음과 같다.

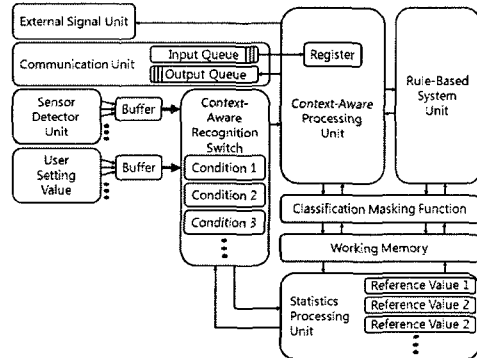


그림 1. 유비쿼터스 컴퓨팅 시스템을 위한 상황 인식 구조

### Sensor Detector Unit:

본 논문에서 제안된 Architecture로 구현된 UoC가 Ubiquitous Computing에 쓰이는 각각의 환경에 따라 다양한 센서를 통해 사용자의 환경 정보를 얻는다. 예를 들면, 사람의 건강에 관한 센싱을 원할 경우 심전도, 근전도, 뇌전도, 온도 등의 정보를 수집할 수 있도록 센싱한다. 또 다른 예로, 건물의 관리를 위한 센싱을 원할 경우 온도, 습도, 형광등의 켜짐과 꺼짐, 문의 열림과 닫힘, 창문의 열림과 닫힘, 사람의 유무 등에 관한 정보를 센싱한다.

### User Setting Value:

이것은 CRS와 매우 밀접한 관련을 맺고 있는 입력 정보이다. 유저가 설정한 Setting Value에 따라 다양한 센서를 계층적으로 분류하여 Level에 따라 각기 다른 출력 결과를 얻을 수 있다. 예를 들어, 사람의 건강 정보를 체크하기 위해 가장 중요한 심장 박동에 관한 정보는 실시간으로 감지할 필요가 있다. 그러므로 심전도에 관한 Setting Value의 비중을 크게 잡고 매우 정밀하게 감시 할 수 있도록 한다. 이와 대조적으로 사람의 생명과는 그다지 밀접하지 않는 인체 정보는 감시 수준의 비중을 작게 두어 Process Unit의 부담을 덜고, 쓸데없는 자료의 수집과 축적을 막을 수 있다.

### Buffer:

센싱된 Data가 실시간으로 Processing Unit로 입력 되면 System에 과부하가 걸린다. 그러므로 어느 정도의 기준을 정해 적당한 간격으로 처리 할 수 있도록 버퍼를 사용한다. 버퍼의 크기가 크면 센싱 Data의 처리가 실시간으로 처리되지 않고, 버퍼의 크기가 작으면 Processor가 처리해야 할 양이 많으므로 적절한 크기를 유지하며 센싱 Data를 Process Unit로 전달해

야 한다.

**Communication Unit:**

Ubiquitous Computing System은 하나의 시스템이 혼자서 상황에 따른 처리를 하는 것이 아니라, 다른 시스템과의 통신을 이루며 여러 정보를 공유하고 협력하는 체제를 이뤄야 한다. 따라서 센서를 통해 입력된 정보를 다른 System과 공유하고 사용자로부터 더욱 최적화된 서비스를 하기 위해 통신을 하는 부분이다.

**Input Queue, Output Queue and Register:**

Communication Unit을 통해 들어온 Data 모두를 Processing Unit로 보내는 것은 System의 과부하를 일으킨다. 그러므로 Queue에 저장하여 먼저 들어온 Data의 처리가 끝나면 순차적으로 처리 할 수 있도록 한다. 또, 통신으로 들어온 Data 전체가 Processing Unit에 필요하진 않을 것이다. 여기엔 통신을 위해 새롭게 인코딩된 Data도 포함되어있을 것이다. 전체 통신 Data중 실질적으로 필요한 Data만을 추출한다.

**CRS (Context Aware Recognition Switch):**

CRS는 UoC Architecture를 위해 본 논문에서 새롭게 제안된 개념으로 각기 다른 다양한 입력의 Data로부터 개별적으로 처리하는 방식이다. 여기에, Cost에 따라 계층적으로 구분된 입력의 처리를 통해 사용자에게 상황에 따른 특화된 서비스를 제공한다.

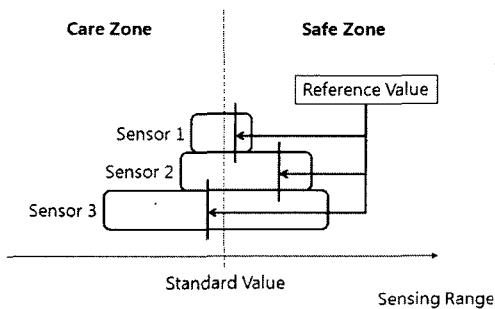


그림 2. CRS: Context Aware Recognition Switch

그림.2는 CRS의 개념을 도식화 한 것이다. 다양한 센서로부터 입력된 Data를 분석하여 안정된 상태인지 그렇지 않으면 특별한 관리 및 처리가 필요한 상태인지 판단한다.

각 센서는 정확한 하나의 값이 입력될 수도 있고, 위 그림과 같이 어느 정도의 변동 폭을 가지질 수도 있다. 이렇게 입력된 정보를 기존의 평균 값과 비교하여 상황을 인지한다. 여기서 중요한 점은 모든 사람의 신체 정보나 주변 환경 정보 등의 상황을 단 하나의 기준 값을 통해 평균 내어 진다면 사용자에게

특화된 서비스를 하기 어렵다. 감시되는 개체가 다양하듯 또, 한 개체가 시간과 장소의 변화 등 처한 상황의 변화에 따라 그에 따른 입력 Data가 다양하게 변화한다. 그러므로 그에 따른 기준 값도 개별적으로 달리 정해져야 한다. 여기서 CRS의 장점이 발휘된다.

사용자로부터 수동적으로 설정된 값이나 혹은 System 스스로가 누적된 Data를 통계화 하여 설정된 값을 기준으로 상황을 개별적으로 체크한다.

또, 다양한 센서의 입력 중에서 중요도의 차이에 따라 계층적으로 서비스를 할 수 있다. Cost Value가 높은 입력을 우선 처리 함으로써 군더더기 없이 실시간 처리가 가능해지고, 각 센서의 세밀함의 및 정확도 등을 달리하여 처리할 수 있다.

**Context Aware Processing Unit:**

CRS를 기준으로 다양하게 인식된 센서 정보를 사용자의 상황에 맞게 서비스를 제공하도록 처리를 하는 부분이다. 입력된 센서 정보를 Working Memory에 저장하고 설정된 CRS를 통해 상황을 인지하고, 그에 따른 특화된 처리를 Rule based System Unit을 통해 결정하여 사용자에게 특화된 서비스를 할 수 있도록 한다.

**Rule Based System Unit:**

사용자 또는 주변 환경 정보를 센서를 통해 입력 받고, 인지된 상황에 따라 최적화된 서비스를 분석해 내는 부분이다. [if ~ then]의 논리적인 분석을 통해 입력된 센싱 정보에 따라 최적화된 서비스를 할 수 있는 출력을 제시한다. CRS에서 설정된 값을 통해 비중이 높은 센싱 정보를 우선적으로 판단한다.

**Working Memory:**

지속적으로 입력된 센싱 정보를 저장하고 관리하는 부분이다. Working Memory에 저장된 센싱 정보는 Statistics Unit을 통해 통계화되어 기준을 정하게 된다. 각각의 센서 별로 개별적인 기준 값을 산출함으로써, 사용자에게 보다 특화된 서비스를 할 수 있도록 한다.

**Classification Masking Function:**

Processing Unit에서 Working Memory에 접근할 때, Working Memory에 있는 모든 Data를 가져올 필요는 없다. 상황에 따라 필요한 Data만을 접근하여 처리 속도를 높이기 위해 필요 없는 Data는 Masking 할 필요가 있다. 이렇게 Classification Masking Unit은 Processing Unit이 Working Memory의 Data에 접근할 때, 상황에 맞는 Data만을 뽑아내기 위해 관계 없는 Data를 Masking하는 기능을 한다.

다음의 그림은 Working Memory와 Classification Masking Function을 도식화 한 것이다.

Address	1	2	3
Data	A	B	C

그림 3. 일반적인 메모리 구조

Address	1	2	3
Data	A	B	
			C

그림 4. UoC를 위해 새로운 메모리 구조 제안

다음의 그림은 Memory의 트리 구조를 나타낸 것이다.

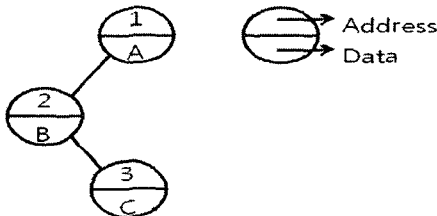


그림 5. 트리 구조의 메모리 구조 제안

제안된 Memory Structure는 트리 구조이다. 그래서 높은 레벨의 센싱 Data의 처리는 하위 레벨의 센싱 Data의 처리에 영향을 준다.

예를 들어, A라는 최상위 레벨의 센싱 Data가 있고, 이 센싱 Data에 특정한 반응이 일어나면, 시스템은 그 하위 레벨의 센싱 Data인 B를 감지 할 필요가 없다.

**Statistics Processing Unit:**

센싱된 환경정보를 통해 사용자의 상황을 판단 할 수 있도록 CRS의 기준 값을 산출하는 부분이다.

**External Signal Unit:**

센서를 통해 인식된 상황 정보에 따라 사용자에게 최적의 서비스를 하기 위해 동작하는 다양한 출력을 가지는 부분이다. 사용자가 특수한 관리, 처리 및 결정이 요구될 때 사용자와 인터페이스를 위한 상황 정보를 표현한다.

**IV. CRS와 통계**

CRS의 기준 값을 정하기 위해 Statistics Unit에서

평균 값을 산출한다. 일반적으로 알려진 평균은 산술 평균, 기하 평균, 조화 평균 등이 있다. 하지만, 이러한 평균 값은 계산 과정이 매우 복잡하다. 이러한 복잡한 계산은 Processor에 무리한 전력 소모를 유발하고, 계산에 따른 처리 시간 또한 길어질 수 밖에 없다.

이것은 실시간으로 처리가 이뤄져야 하는 Ubiquitous Computing System에 적합하지 않은 방법이다. 또, 일반적으로 알려진 평균들은 그 값을 계산해내기 위해 전체 Data 값을 기억하고 있어야 한다. 그러므로 Data의 양이 많아질수록 Memory에 부담이 커지게 된다.

본 논문은 이러한 문제들을 해결하기 위해 다음의 통계적 사용자 상황적 분포 값을 이용하여 CRS값을 결정하는 방법을 선택하게 된다.

$$Avr(N) = (1 - W) * Avr(N - 1) + W * N \quad (1)$$

Table.1 Notations define

Avr(N)	Average until present Just like general average from 1 to n. $\frac{1}{N} \sum_{i=1}^n X_i \quad (2)$ But, Avr(N) is better simple.
Avr(N-1)	Average until former Just like general average from 1 to n-1. $\frac{1}{N} \sum_{i=1}^{n-1} X_i \quad (3)$ But, Avr(N-1) is better simple.
N	Present sensing data
W	Weight value

본 논문에서 제시하는 평균의 수식(1)은 이전의 평균(2)들과 비교하여 매우 간단하다. 따라서, 계산 과정에서 처리에 따른 전력 사용이 비교적 적고, Ubiquitous Computing System에 활용할 수 있도록 빠르게 계산할 수 있다. 또 본 논문에서 제시하는 평균의 수식(1)은 이전까지의 평균 값만을 Working Memory에 저장하여 Memory의 과중한 업무를 최소화 할 수 있다.

마지막으로 결정적으로 중요한 이점으로는, Weight Value를 통해 CRS의 비중을 다양하게 정할 수 있다는 것이다. Weight Value가 크면 현재 센싱된 정보의 신뢰도를 높여 즉각적인 반응을 할 수 있도록 처리한다.

반대로 Weight Value가 작으면 현재 입력된 센

심 값보다 기존의 평균값을 더 신뢰한다. 이는 지속적인 관찰을 요하는 환경에서 사용 될 수 있다. 다음의 표를 통해 각 Weight Value에 따라 달라지는 평균 값을 볼 수 있다.

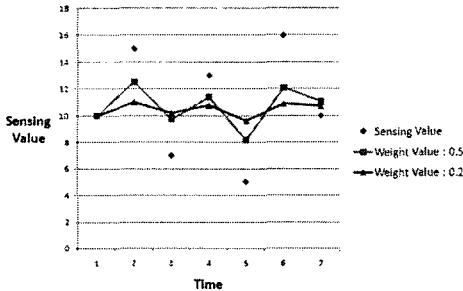


Fig. 3. Weight Value에 따른 통계 값

Fig.3 과 같이 현재 입력된 값이 평균 값에 미치는 영향력을 정해주는 Level을 두어 다양한 정도의 출력을 할 수 있다. Sensing Value는 현재 입력된 센싱 Data이다.

여기에 Weight Value: 0.5 와 Weight Value: 0.2의 평균 값을 산출하였다. Weight Value: 0.5은 현재 입력된 센싱 Data 값의 비중을 높이 설정 한 것이고, Weight Value: 0.2는 센싱 Data 값의 비중을 낮게 설정 한 것이다. Weight Value: 0.5은 기존의 평균 값과 비교하여 현재 입력된 정보의 영향력이 커져 평균 값의 변화율이 크다. 하지만, Weight Value: 0.2는 현재 입력된 정보의 영향력이 작아 평균 값의 변화율이 작다.

Weight Value의 값에 따라서 현재의 상황에 민감하게 반응할 수도 있고, 천천히 상황에 대한 변화를 지속적으로 감지하여 서비스할 수 있는 것이다.

### V. 결론

본 논문은 CRS의 개념을 도입하여 UoC에 활용할 수 있도록 새로운 Architecture를 제시하였다. 또한, CRS의 Reference Value를 산출하기 위해 간단 하면서, 특화된 서비스를 할 수 있도록 Weight Value를 사용한 평균 값의 산출 과정을 보여주었다.

CRS를 활용한 상황인식 기반의 UoC Architecture는 다양한 분야에 활용될 수 있다. 예를 들어, U Health Care System에서는 사용자의 생체 정보를 다양한 센서를 통해 인식하고, 각 상황에 따라 최적의 방법으로 사용자를 인식하고 처리하게 된다. 또, U Home System에서는 집안의 다양한 정보들을 센서를 통해 인식하고, 집안의 여러 장비들을 원격으로 제어하거나 자동적으로 온도나 습도를 조절하는 등의 서비스를 제공할 수

있다. 이처럼 Sensor Network을 형성할 수 있는 다양한 곳에서 응용이 가능하다.

### References

- [1] Jung Heon Man, Lee Jung Hyun "Probability annotated Ontology Model for Context Awareness in Ubiquitous Computing Environment," the Republic of Korea, 2006. 7.
- [2] Joseph Giarratano, Gary Riley "EXPERT SYSTEM Principles and Programming," PWS KENT Publishing Company pp. 501 532, 1989.
- [3] Mark Weiser. "The Computer for the Twenty First Century." Scientific American. pp. 94 101. September 1991.
- [4] Mark Weiser, <http://www.ubiq.com/hypertext/weiser/UbiHome.html>.
- [5] G.D. Abowd, A.K. Dey "Towards a Better Understanding of Context and Context Awareness," Proc. 1st Int'l Symp. Handheld and Ubiquitous Computing(HUC 99), Lecture Notes in Computer Science, no 1707 Springer Verlag, Germany, pp. 304 307, 1999.
- [6] Abowd, G.D.; Ebling, M.; Hung, G.; Hui Lei; Gellersen, H. W. "Context aware computing" Pervasive Computing, IEEE , Volume: 1 , Issue: 3 , July Sept. pp. 22 23, 2002.
- [7] Munoz, M.A.; Rodriguez, M.; Favela, J.; Martinez Garcia, A.I.; Gonzalez, V.M. "Context aware mobile communication in hospitals," Computer , Volume: 36 , Issue: 9 , Sept. pp. 38 46, 2003.
- [8] B. Yoshimi. "On Sensor Frameworks for Pervasive System." Proc. Of Workshop on Software Engineering for Wearable and Pervasive Computing, June 2000.
- [9] Aleksandar Milenković, Chris Otto, Emil Jovanov "Wireless Sensor Networks for Personal Health Monitoring: Issues and an Implementation," Elsevier Journal of Computer Communications, August 2006.