
Ubiquitous Computing을 위한 CRS와 DOS 기반의 Context-Aware System Architecture

두경민, 김순국, 이강환

CRS and DOS based Context-Aware System architecture for Ubiquitous Computing

Kyoungmin Doo, Sunguk Kim, Kangwhan Lee

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

요 약

현재 기술 동향에 따라, 언제 어디서나 통신 및 컴퓨팅이 가능한 유비쿼터스 컴퓨팅 시스템(Ubiquitous Computing System)을 위해 사용자 및 주변 환경의 상황을 인식하는 기술(Context-Aware Computing System)이 필수적인 요소로 부각되고 있다. 하지만, Context-Aware Computing System을 구현하기 위해서 사용자 및 주변 환경으로부터 입력되는 불확실하거나 모호한 상황정보에 대한 표현과 추론에 대한 연구는 부족한 실정이다. 이를 해결하기 위해 Rule-based System을 기반으로 CRS와 DOS의 개념을 도입하여 상황인식 기술을 SoC로 구현할 수 있도록 새로운 Architecture를 제안한다. 마지막으로 CRS와 DOS가 Network Topology에 적용된 RODMRP(Resilient Ontology-based Dynamic Multicast Routing Protocol)의 예를 통해 그 효용성을 입증하고, Ubiquitous Computing System에서 활용 가능한 방법 등을 제시한다.

Key World:

Ubiquitous Computing System, Context-Aware, Context Recognition Switch, Dynamic Optimal Standard

I. 서론

최근 컴퓨터의 개념이 정보 산업과 이동 통신 기술이 발전함에 따라 퍼스널 컴퓨터를 넘어 매우 빠르게 확장되고 있다. 이동 정보단말 형태의 PDA, HPC(Hand-Held PC), 휴대폰 등의 휴대용 기기나 정보 가전제품 등이 새롭게 확장되면서 컴퓨터 시스템들은 인간 생활의 편리성과 다양성을 위해 하나의 네트워크에 연결되고 있다. 이런 상황에서 언제 어디서나 통신 및 컴퓨팅이 가능하고 컴퓨팅 시스템이 상호간에 정보를 공유하고 협력함으로써 새로운 서비스가 사용자에게 제공이 가능한 형태의 Ubiquitous Computing 환경으로 발전하고 있다[1, 2].

Ubiquitous Computing이란, 사용자가 컴퓨터나 네트워크를 의식하지 않고 장소에 상관없이 자유롭게 네트워크에 접속할 수 있는 환경을 말한다.

Ubiquitous Computing System을 구현하기 위해 기존의 컴퓨팅 환경과 같이 사용자와 컴퓨터간의 대화형 상호작용이 아닌 물리적인 환경 상황(Context)등을 시스템이 스스로 인식하고 이를 기반으로 사용자와의 상호 작용을 지원하는 상황 인식 기술이 필수적인 요소로 부각되고 있다. 이런 이유에서 Context-Aware Computing System에 관한 연구가 활발히 진행되고 있다[3].

Context-Aware Computing System은 주변 환경을 여러 센서를 통해 감지하여 Context를 파악 및 인지하고, 그 인식된 정보에 의해 최적화된 서비스를 제공하는 시스템이다[4, 5]. Context-Aware Computing System을 제공하기 위해 무엇보다도 사용자 정보 및 주변 환경 정보를 감지하는 센서(Sensor) 기술이 필요하다. 기존의 키보드나 마우스 등의 컴퓨터 인터페이스를 극복하기 위해 사용자의 표정, 동작, 음성, 생체 인신호 등의

사용자 중심 인터페이스 기술이 필요하다. 하지만, 사용자 및 주변 환경으로부터 입력되는 불확실하거나 모호한 상황정보에 대한 표현과 추론에 대한 연구는 부족한 실정이다 [6].

본 논문은 차세대의 Ubiquitous Computing System에서 이러한 문제를 해결하기 위해 이미 개발된 Rule-based System을 기반으로 CRS(Context Recognition Switch)와 DOS(Dynamic and Optimal Standard)라는 새로운 개념을 도입한 Context-Aware Computing System Architecture를 제안한다.

Rule-based System은 인공지능 분야에서 성공적으로 실제 응용 분야에 적용된 기법으로 그 우수성이 잘 알려져 있다[7]. 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 본 논문은, Rule-based System에 CRS와 DOS를 도입하여 효율적인 실시간 처리가 가능한 새로운 Architecture를 제안하고 있다.

본 논문의 구성은 다음과 같다. 제 2장에서는 Rule-based System의 실시간 처리를 위해 도입된 CRS와 DOS의 개념을 설명하고, 3장에서는 CRS와 DOS의 개념이 도입된 Context-Aware Computing의 UoC(Ubiquitous on Chip)로의 개발을 위한 새로운 Architecture를 소개한다. 4장에서는 본 논문이 제안하는 Architecture가 Network Topology에서 활용되고 있는 RODMRP의 예를 들어 그 효율성을 증명하고, 마지막 결론에서는 Ubiquitous Computing System에서 본 논문에서 제안 적용된 CRS와 DOS의 특징과 활용 방법에 대해 보여준 후, Ubiquitous Computing System의 향후 발전 가능성을 예상해봄으로써 본 논문에서 제시한 Context-Aware Computing System Architecture의 성능을 평가한다.

II. 본 논문이 제안하는 CRS와 DOS의 정의

Ubiquitous Computing System에서 사람에게 보다 최적화된 서비스를 제공하기 위해 사람의 상태 및 주위 환경에 해당하는 매우 다양한 Context를 인지해야 한다. Ubiquitous Computing System은 다양한 Sensor를 통해 입력된 Context에 따라

미리 정해놓은 기준으로 어떤 서비스를 제공할 것인지 판단하게 된다. 여기서 기존의 고정적인 (UNIFORM) 기준 값은 시시때때로 변화하는 사람과 환경의 다양한 상황에 대해 최적화된 서비스를 판단하기에 어려움이 있다. 따라서, Context의 패턴을 인식하여 유동적인 기준 값을 제시함으로써, 다양한 상황에 유연하게 대처할 수 있는 CRS와 DOS를 제안한다. 여기서 말하는 CRS란 다양한 Sensor를 통해 입력된 값에 가중치를 두어 중요한 Context를 감지하는 Sensor에는 민감하게 반응할 수 있고, 그다지 중요하지 않은 Context를 감지하는 Sensor는 감지 효율을 낮춰 System에 부하를 덜 수 있도록 도와주는 처리이다. 또, DOS는 하나의 입력된 값에 따라 Care상태인지 Safe 상태인지 판단하는 기준을 사람과 주위 환경의 패턴에 따라 변화시켜 각각의 개인에게 최적화된 서비스를 제공할 수 있도록 하는 상황 판단의 기준이 되는 값이다. 이것은 현재 입력된 값과 기존에 저장되어있는 기준 값의 평균 계산을 통해 구해지게 된다. 이때, 그 동안 입력된 모든 Data를 저장하는 것이 아니라 지금까지 계산된 결과 기준 값만을 저장하는 방식을 택함으로써 메모리 효율을 더욱 높일 수 있고, 이는 Ubiquitous Computing System에 매우 효과적으로 활용된다.

III. 유비쿼터스 컴퓨팅 시스템을 위한 상황 인식 구조 제안

본 논문은, 상황인식 기술을 SoC(System on Chip)로 구현하여 Ubiquitous Computing System을 실현하고자 새로운 UoC Architecture를 제안한다. Ubiquitous Computing System을 위해 제안된 Architecture는 다음과 같다.

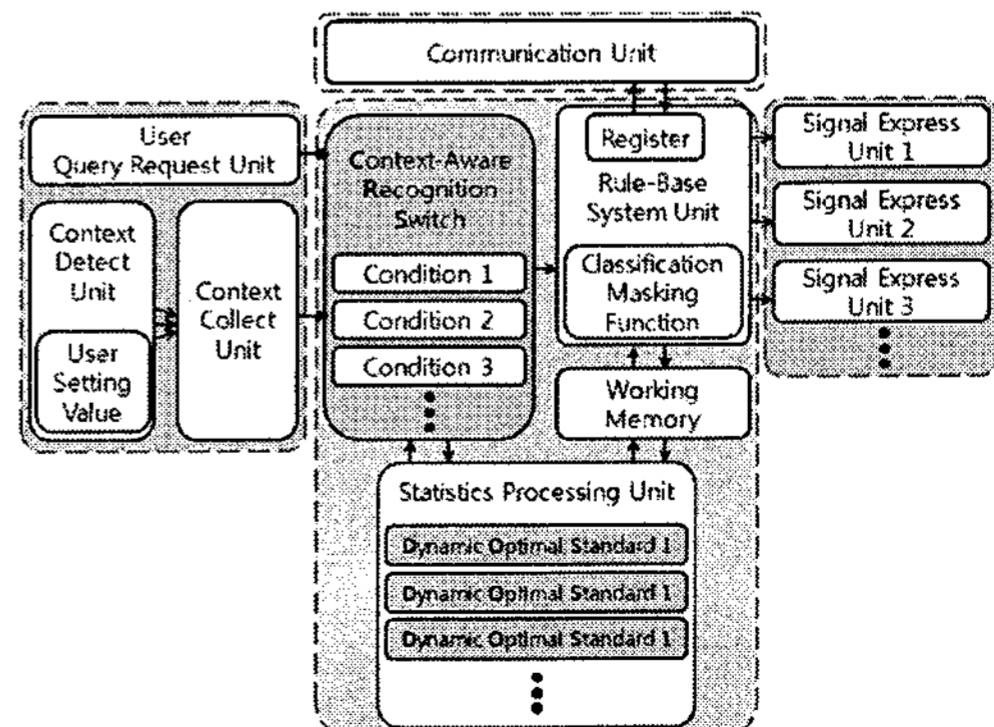


그림 1. 유비쿼터스 컴퓨팅 시스템을 위한 상황 인식 구조

전체적인 구조는 크게 4개의 파트로 구성된다. 그 첫 번째는, 사용자 및 주변 환경에 대한 상황을 인지하기 위해 다양한 Sensor를 통해 정보를 수집하는 Context Detect 및 Context Collect 부분이다. 다양한 멀티 센서를 통해 보다 자세한 정보를 수집하는 역할을 Context Detect Unit이 한다면, Context Collect Unit은 수집된 정보 중 실질적으로 System에서 요구하는 데이터를 분별하는 역할을 한다. 또한, 이처럼 System 스스로가 지능적으로 Context를 수집할 수도 있지만, 경우에 따라서는 User Setting Value를 통해 사용자가 System을 직접적으로 제어할 수 있다.

두 번째 파트는 다양한 센서들을 통해 수집된 정보를 가공 처리하는 부분으로 Context-Aware Computing System Architecture에서 가장 핵심적인 부분이다. 우선 Context 수집단계에서 넘어온 다양한 정보들을 모두 처리하기에는 System의 과부하를 일으킨다. 이에 따라, 각 상황을 판단하여 중요도가 높은 정보를 우선 선택 및 처리함으로써 보다 빠르고 안정된 처리를 도울 수 있다. 이것이 CRS에서 하는 중요한 역할이다.

CRS에서 우선 처리해야 할 정보를 선택한 후에 이를 Rule-based System Unit에 전달하여 사용자에게 최적의 서비스를 판단하게 된다. 이때 사용자의 상황을 판단하는 기준인 DOS가 적용된다. 일반적인 System에서 입력에 따른 출력의 판단 기준은 고정적인 것이 대부분이다. 하지만 이것은 다양한 사용자의 특별화된 서비스를 제공하기 힘들다. Ubiquitous Computing 환경에서는 개 개인의 각 특성의 차이점을 분명하게 판단하여, 그 상황에 따라 최적화된 서비스를 제공해야 한다. 이를 위해 사용자 및 환경의 상황을 판단하는 기준은 정해진 하나의 값이 아니라 각 상황에 따라 유동적으로 변화하는 값이 되어야 하겠다. 이것이 사용자 및 환경 변화에 따라 유동적으로 변화하면서 최적화된 서비스를 제공하기 위한 판단 기준이 되는 DOS의 개념이다. 그 계산 과정은 다음과 같다.

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

- Avg(N): 현재의 판단 기준 값
- Avg(N-1): 이전까지의 기준 값
- W: Weight value
- N: Sensor로 부터 입력된 현재 입력 값

세 번째 파트는 제안한 Architecture의 처리 결과를 표현할 수 있는 출력 파트이다. Context-

Aware Computing System에서 처리된 최적화된 서비스를 사용자에게 제공하는 파트이다.

네 번째 파트는 제안한 Architecture의 처리 결과를 다른 시스템과 공유하기 위한 무선 통신 인터페이스이다. 이 부분은 다음 장에서 조금 더 깊은 설명이 이뤄질 것이다.

다음의 그림2에서는 Context-Aware Computing System의 처리 과정을 설명한 UoC의 구조를 보여준다.

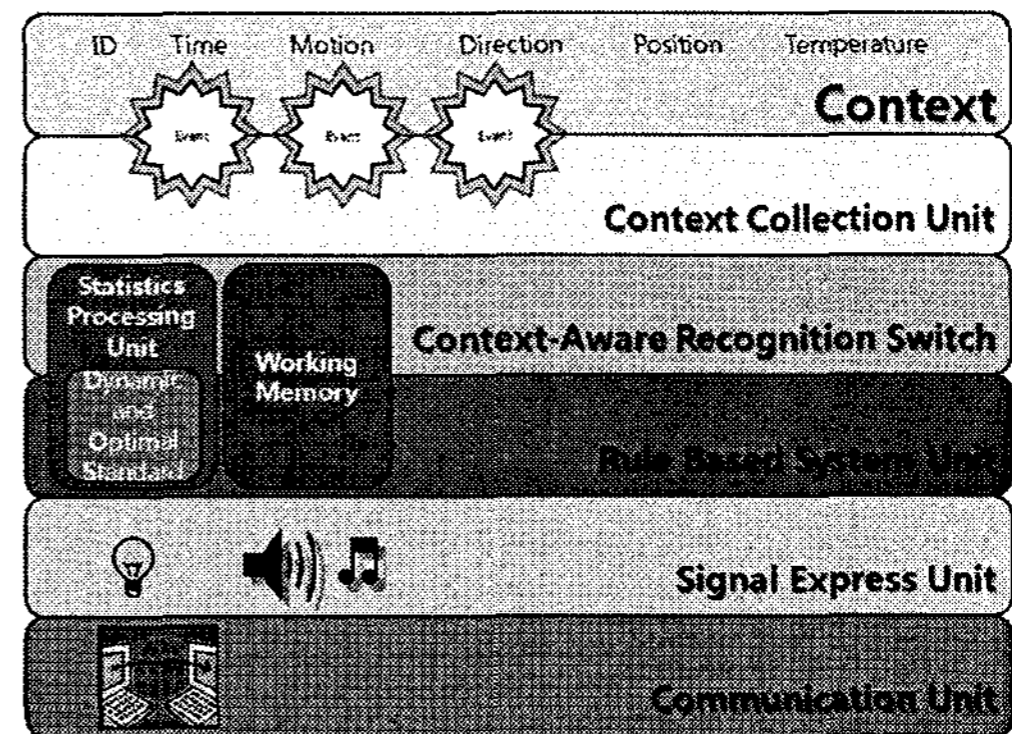


그림 2. UoC Architecture

사용자 및 환경에 대한 다양한 Context를 Sensor를 통해 인식한다. 인식된 상황 정보를 Context Collection Unit을 통해 불필요한 데이터를 검출한 후, Rule-Based System에 전달하여 최적의 서비스를 취할 수 있도록 처리한다. 이때, Context-Aware Recognition Switch는 다양한 Sensor를 통해 들어온 데이터에 대해 가중치를 두어 중요도가 높은 상황을 우선적으로 처리할 수 있도록 한다. 또한, 사용자의 상태가 Safe 상태인지, Care 상태인지 판단하기 위해 DOS를 사용한다. 마지막으로 Context-Aware Computing System에서 사용자의 최적의 서비스 제공하기 위해, 다양한 출력 디바이스에 해당하는 Signal Express Unit이 사용되고, Communication Unit을 통해 각 System의 정보를 다른 System과 공유할 수 있다.

IV. Ad-Hoc Network의 Topology에서 활용된 CRS와 DOS

Ad-Hoc Network는 노드(Node)들에 의해 자율적으로 구성되는 기반 구조가 없는 네트워크를 말한다. Ad-Hoc Node들은 무선 인터페이스를 사용하여 서로 통신하고, 멀티 홉 라우팅 기능에 의해 무선 인터페이스가 가지는 통신 거리상의 제약

을 극복하며, Node들의 이동이 자유롭기 때문에 네트워크 Topology가 동적으로 변화되는 특징이 있다. Ad-Hoc Network 를 형성하기 위한 Topology 중에서 이동 Node간의 변화된 환경에서 복원력이 뛰어난 트리 구조의 효율적인 구조를 가지는 것이 RODMRP이다[8].

RODMRP의 Node들간의 계층적인 구조는 다음과 같다.

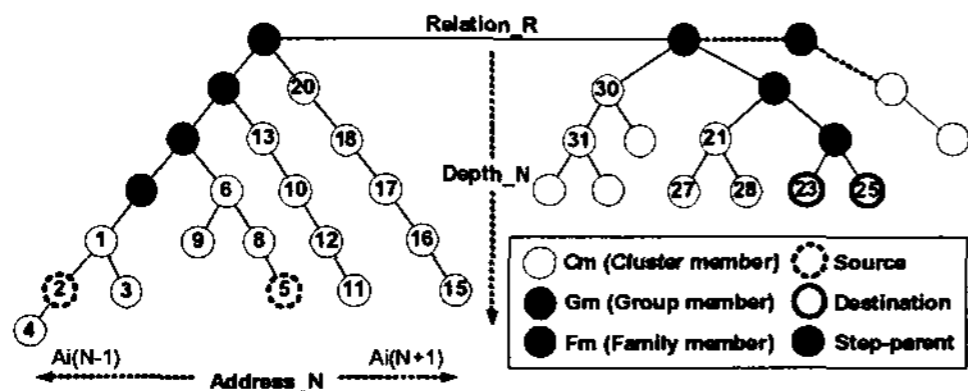


그림 3. RODMRP에서 계층적인 트리 구조를 가지는 Node

RODMRP에서 계층적인 Node의 Network Topology를 형성하기 위해서는, 각 Node들간의 비교를 통해 최상의 컨디션을 갖는 Node가 최상위의 Node가 되어 다른 하위 Node를 관리 하게 된다. 이렇게 Network Topology를 형성하는 과정에서 각 Node의 컨디션을 판단하기 위해 고정적인 기준을 사용한다면, 그 비교 과정에서 많은 연산을 필요로 한다. 하지만, DOS의 개념을 도입함으로써 최상위 Node가 되었던 Node는 주로 최상위 Node가 되도록 처리하고, 하위 Node가 되었던 Node는 주로 하위 Node가 되도록 처리하여, 그 비교 처리 과정의 업무를 효율적으로 관리 할 수 있다. DOS 계산 과정 및 결과는 다음과 같다.

$$DOS(N) = (1 - W) * DOS(N-1) + W * N \quad (2)$$

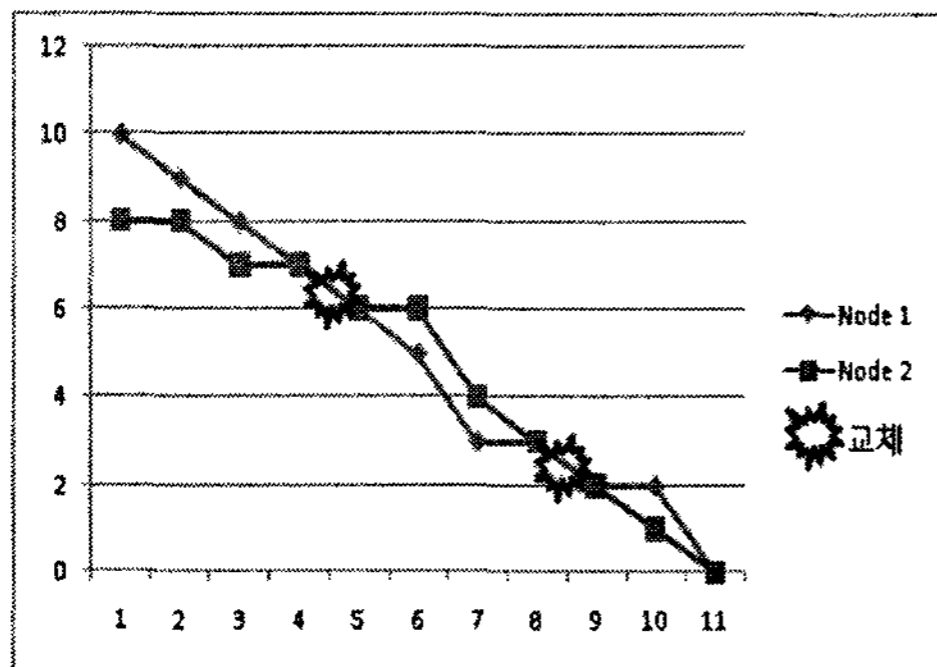


그림 4. 일반적인 RODMRP에서 상위 Node와 하위 Node의 교체

- 1번 Node의 초기 에너지 상태는 10이라 놓고, 2번 Node의 초기 에너지 상태는 8이라 설정한다.
- 에너지가 높은 Node가 상위 계층의 Node가 되어 하위 Node를 관리 한다.
- 상위 계층의 Node는 처리량이 많아 에너지가 2씩 떨어진다.
- 하위 Node는 처리량이 적어 에너지가 1씩 떨어진다.
- 상위 Node와 하위 Node의 에너지 상태가 역전되면 이들의 관계를 교체한다. 이 과정에서 데이터를 서로 주고 받는 복잡한 처리를 하기 때문에 에너지가 2를 소비한다.

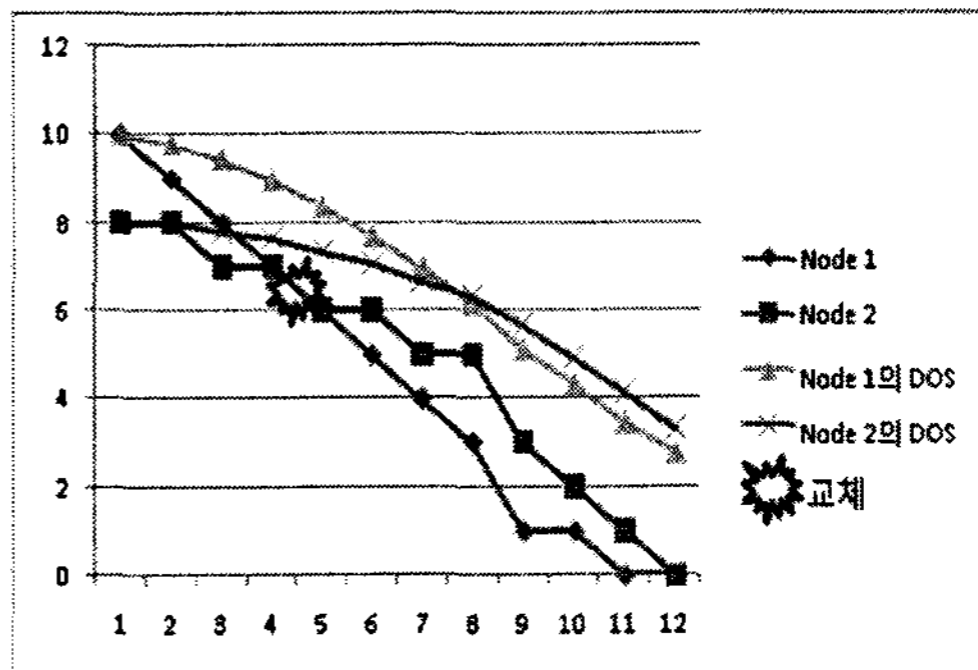


그림 5. DOS를 활용한 RODMRP에서 상위 Node와 하위 Node의 교체

- 상위 Node와 하위 Node의 에너지 상태가 역전되어도 바로 그 역할을 바꾸는 것이 아니라, DOS값이 역전되는 최종적인 상황에서 Node들의 관계를 교체한다.
- 위 그림에서 DOS의 Weight Value의 값은 0.1로 설정되어있다. 즉, 현재 입력된 Sensor 값보다 이전에 계산된 DOS의 값의 비중을 더 높여 계산한 결과이다.

RODMRP에서 상위(Family or Cluster) Node와 하위(Member) Node의 에너지 상태가 역전이 되면 그 역할을 바꾸게 된다. 여기서 많은 데이터를 주고 받아야 하기 때문에 많은 에너지 소비를 유발하게 됩니다. 그러므로, 상위 Node와 하위 Node의 역할 교체를 최대한 줄여야 합니다. 이를 위해 DOS를 활용하여, 처음 인식된 상위 Node는 큰 변화가 없을 때까지 지속적으로 상위 Node의 역할을 지정 하고, 하위 Node 역시 큰 변화가 없을 때까지 지속적으로 하위 Node의 역할을 지정 할 수 있다.

결론적으로 상위 Node와 하위 Node의 관계를 교체하는 횟수가 줄어들어 Network를 구성하기 위한 복잡한 처리 과정을 피할 수 있다. 이로써 전체적인 Node의 Life Time이 길어지는 것을 확인할 수 있다.

V. 결론

본 논문은 CRS와 DOS의 개념을 도입하여 UoC에 활용할 수 있도록 새로운 Architecture를 제시하였다. 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 Auto System에서는 집안의 다양한 정보들을 센서를 통해 인식하고, 집안의 여러 장비들을 원격으로 제어하거나 자동적으로 온도나 습도를 조절하는 등의 서비스를 제공할 수 있다. 이처럼 Sensor Network를 형성할 수 있는 다양한 곳에서 응용이 가능하다.

앞으로 UoC Architecture의 연구는 보다 세부적인 구조로 나아가야 할 것이고, 사용자 주변 환경의 상황을 감지할 수 있는 다양한 센서 기술이 개발되어야 할 것이다. 마지막으로 최적화된 서비스를 사용자에게 제공하기 위해 사용자 중심의 인터페이스 기술이 개발되어야 할 것이다.

References

- [1] Mark Weiser. "The Computer for the Twenty-First Century." *Scientific American*. pp. 94-101. September 1991.
- [2] Mark Weiser, <http://www.ubiq.com/hypertext/weiser/UbiHome.html>.
- [3] Jung Heon Man, Lee Jung Hyun "Probability-annotated Ontology Model for Context Awareness in Ubiquitous Computing Environment," the Republic of Korea, 2006. 7.
- [4] 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.
- [5] 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.
- [6] Kyoungmin Doo. "A study on the Context-Aware Architecture for Ubiquitous on Computing System," The Korea Institute of Maritime Information and Communication Sciences. 2007.
- [7] Joseph Giarratano, Gary Riley "EXPERT SYSTEM Principles and Programming," PWS-KENT Publishing Company pp. 501-532, 1989.
- [8] Sun-guk Kim. "A study on Inference Network Based on the Resilient Ontology-based Dynamic Multicast Routing Protocol," The Korea Institute of Maritime Information and Communication Sciences. 2007.