











**상황의 연속성** - 일반적으로 사람의 행동 및 환경의 변화는 일시적으로 무분별하게 일어나는 것이 아니라 일정 시간 동안 지속된 후 일어나게 된다. 예를 들어, 사람이 이동을 위해 걷는 행동은 목적지에 도착할 때까지 지속되며, 지속 시간은 목적지까지의 거리에 따라 달라지게 된다. 이러한 관찰에 기반하여, 현재 사용자의 행동 및 상황에 맞게 센서를 재구성했을 경우 상황이 계속 되는 기간 동안 관련이 적은 센서의 정확도를 낮추어도 전체적인 상황정보 수집에 큰 영향을 주지 않으며, 소모되는 에너지를 줄일 수 있다.

**상황의 패턴성** - 사용자의 행동이나 환경의 변화는 특정 순서에 따라 일어나게 되고, 이는 시간과 매우 밀접한 관련을 가지고 있다. 이는 사용자의 생활 패턴에 기인한 것으로, 사용자의 직업에 따라 특정한 패턴을 가지고 일어나게 된다. 예를 들어, 일반적인 직장인은 출근, 업무, 점심, 업무, 퇴근의 특정 행동 패턴을 가지게 된다. 이러한 패턴을 통해서 특정 시간에 일어나는 사용자 행동의 지속시간을 과거 기록을 통해 대략적으로 예측할 수 있다.

이 두 가지 관찰 결과를 바탕으로, 사용자와 관련이 있는 행동 및 상황을 분석하여 관련 높은 센서의 정확도는 유지하며 관련이 적은 센서의 성능을 낮춰 소모 에너지를 줄이는 것이 동적 센서 재구성 기법의 핵심 개념이다. 그림은 센서와 센서 설정 인자들 사이에 관계를 도식화 한 것으로, 샘플링 레이트는 사용자 상태(User State)에 기반하여 동적으로 재구성되며, 센서 활성화 주기는 과거 기록을 바탕으로 재구성된다.

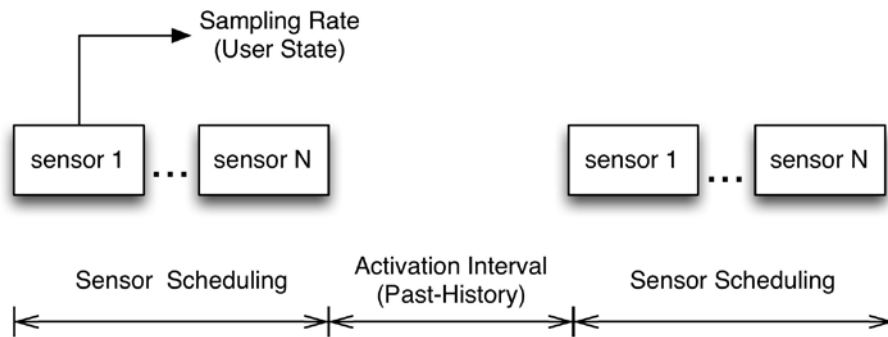


그림 3 센서 재구성 기법과 센서 설정 인자 사이의 관계

## 5.1. 사용자 상태 기반 센서 재구성 기법

사용자가 특정 행동 및 상황에 처한 경우, 연속성을 가지는 센서들의 정보를 추출하기 위해 사용자 상태 명세 양식(User State Specification Template)을 이용한다. 사용자 상태는 사용자 상황정보들의 집합으로 표현이 되며, 사용자 상태 명세 양식에는 특정 사용자 상태가 감지될 경우 센서를 재구성하는 방법에 대해서 서술 한다. 재구성된 센서의 설정은 사용자 상태가 변경될 때까지 계속 지속된다. 표 4 는 사용자 상태 명세 양식의 예제를 보여준다. 이 예제는 “meeting” 이라는 유저 상태를 정의 하는 하는 예제로써, 사용자 상황정보 중 “stay”, “speech” 그리고 “located at the office” 의 세 가지 상황정보 조건을 가지고 있다. 따라서 주어진 세 가지 상황정보가 상황정보 수집 과정을 통해 감지되었을 경우 현재 사용자는 “meeting” 상태에 있다고 판단하게 된다. “meeting” 상태가 감지가 되었을 경우, 사용자 상태 명세 양식에 같이 명세 된 센서 재구성 방법에 따라 센서의 설정을 동적으로 변경하게 된다. 주어진 예제에서는 “meeting” 상태가 감지되었을 경우 가속도계의 샘플링 레이트를 낮추도록 명시하고 있다. 이러한 명세를 통하여 사용자 상태를 명세할 수 있으며, 이에 따른 센서의 재구성을 통해 특정 상황과 연관이 적은 센서의 성능을 낮춤으로써 불필요한 에너지 소모를 줄일 수 있다.

표 4 사용자 상태 명세 예시

<pre> User state :   name : "meeting",   condition context : "stay",   condition context : "speech",   condition context : "located at the office", sensor reconfiguration :   user state : "meeting",   sensor : "Accelerometer",   sampling rate : "low" </pre>
---

## 5.2. 과거 기록 기반 센서 재구성 기법

앞서 설명하였듯이, 사용자가 특정 행동을 하거나 주변 환경이 변화한 후에는 어느 정도의 시간 동안 수집되는 상황정보에 큰 변화가 없는 안정화된 상태를 유지하게 된다. 하지만, 사용자의 상태 변화가 일어나게 되면 수집되는 상황정보는 큰 변화를 보이게 된다. 이러한 변화에 효율적으로 대처하기 위해서는, 사용자 상태 변화가 일어나는 시점을 예측하여 상태 변화를 측정(?)할 필요성이 있다. 이러한 변화를 예측하기 위해, 우리는 사용자의 생활 패턴을 기록하여 이를 이용하는 방법을 제안한다. 본 연구에서 사용자 생활 패턴을 기록하기 위해, 하루를 30 분씩 48 개의 슬롯으로 나눠 기록하는 방법을 사용한다. 만약 특정 슬롯에서 평균 보다 높은 사용자 상태 변화가 기록되었다면, 센서 활성화 주기를 낮게 함으로써 보다 민첩하게 사용자 상태 변화를 감지 할 수 있도록 하는 방법을 제안한다. 사용자의 상태 변화를 기록함으로써, 사용자가 활발한 시간과 그렇지 않은 시간을 구분 지어 활발한 시간에는 보다 자주 상황 정보를 수집하여 보다 높은 정확도를 가지게 된다. 반대의 경우에는 상황 정보를 수집하는 횟수를 줄임으로써 에너지를 절약할 수 있다. 센서 활성화 주기는 수식 2 를 통하여 계산된다.  $D(i)$ 는  $i$  번째 시간 슬롯에서 센서 활성화 주기,  $N(i)$ 는  $i$  번째 슬롯에 기록된 사용자 상태변화 횟수,  $\alpha$ 는 전체 평균을 센서 활성화 주기의 기본값으로 맞추기 위한 변수,  $D_{default}$ 는 기본 센서 활성화 주기를 의미한다.

수식 2 센서 활성화 주기 계산식

$$D(i) = \left( \frac{Mean(N(1), \dots, N(48))}{N(i)} + \alpha \right) \cdot D_{default}$$

## 6. 실험 및 검증

### 6.1. 구현 개요

제안된 방법을 구현하기 위해서 스마트폰 상에서 작동 가능한 시스템의 아키텍처가 그림 4 의 형태로 개발되었다. 제안된 아키텍처는 실제 삼성 Galaxy Nexus 스마트폰에 구현 되었으며, 안드로이드 버전 4.1.1 과 커널 버전 3.031 을 사용하고 있다. 각각의 물리적 또는 논리적 센서들은 Sensor 를 상속받아 선언되어있는 네 가지 함수를 필수적으로 구현하여야 한다. 네 가지 함수는 센서의 초기화(init), 센서 작동(resume), 센서 작동 중지(pause), 그리고 센서 값 획득(read)로 구성되고 각각의 함수들 실행에 필요한 정보들은 인자로써 받게 된다. Sensor Manager 는 센서들의 제어에 관련된 관리 및 분석기(Analyzer)와 센서들을 연결시켜주는 기능을 한다. 분석기는 센서로부터 값을 획득하여 수학적 또는 논리적 연산을 통하여 상황정보 판단에 필요한 정보 형태로 처리하게 된다. 분석기의 대표적인 예로는 평균, 분산, 도함수 등의 계산이 있다. 관리 전략(Management Strategy)은 센서들의 관리를 위한 클래스로써 센서의 스케줄링과 재구성에 관련된 정보들을 처리한다. 관리 전략 클래스에서는 다수의 하위 센서 관리 전략을 허용하므로 다양한 센서 관리 방법을 복수로 구현할 수 있다. 앞서 설명한 사용자의 현재 상태에 기반한 센서 관리 기법이 센서 관리의 하위 클래스로 구현된다. 마지막으로, 상황정보 관리자(Context Manager)는 수집되는 상황정보 전반에 관련된 기능을 수행한다. 상황정보 관리기는 분석기로부터 센서 값을 분석한 결과 값을 받아와 현재 사용자의 상황정보를 판단한다. 추가적인 구성요소로써 각 구성요소에서 필요한 정보를 파일 또는 데이터 베이스에 저장하는 기록기(Logger)와 외부 서버에 기록된 정보를 전송할 수 있는 업로더(Uploader)가 있다.

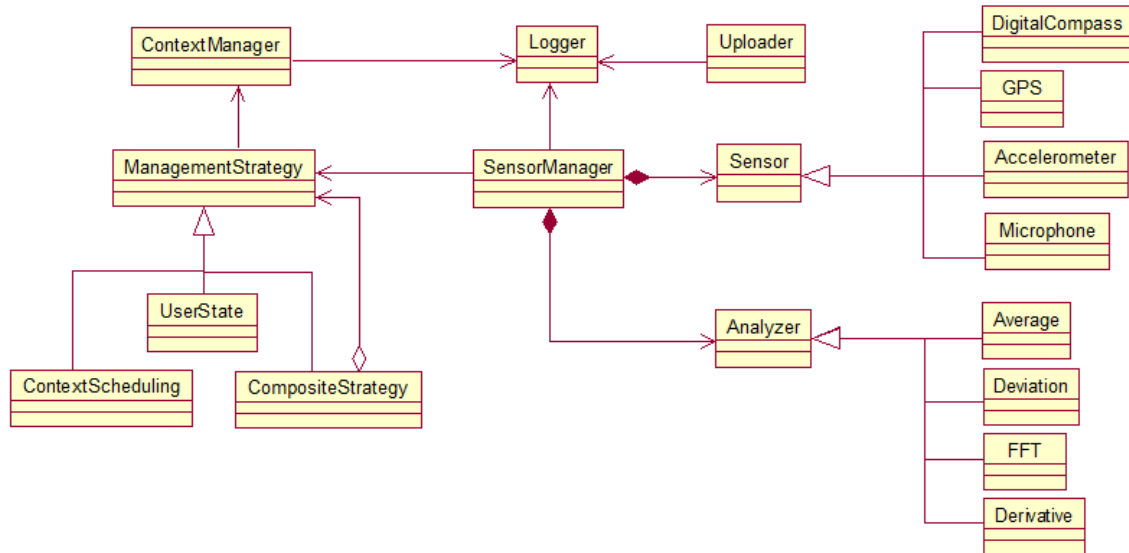


그림 4 제안된 시스템 아키텍처 다이어그램

## 6.2. 센서 설정 인자(Sensor Configuration Parameter) 구성

본 연구에서 기본 센서 활성화 주기는 90 초로 설정된다. 이후, 센서 스케줄링을 위해 3 장에서 사용하는 수식에서 센서의 샘플링 레이트  $SR(S_j)$ 와 에너지 소모량  $ES(S_j)$ 를 추출할 필요가 있다. 실험의 단순화를 위해, 우리는 센서의 샘플링 레이트를 자유롭게 설정가능한 것이 아닌 일반 및 저성능의 두 단계로 변경 가능하도록 설계하였다.  $ES(S_j)$ 값을 추출하기 위해서, 안드로이드에서 기본으로 제공하는 Battery API를 이용하여 각 센서의 에너지 소모량을 측정하였다.  $ES(S_j)$ 의 측정 시, 필요한 것은 센서들 사이의 에너지 소비 비율이므로 정확한 물리적 단위를 사용하여 특정하는 것이 아니라 소비 비율만을 측정하였다. 측정 결과는 표 5 와 같다. 특히, GPS의 경우 샘플링 레이트를 변경하는 것을 안드로이드에서 허용하지 않기 때문에 별도로 샘플링 레이트를 바꾸지는 않는다.

표 5 센서 설정 인자

인자	SR()	ES()
가속도계 - 저성능	5 Hz	1
가속도계 - 일반	16 Hz	1.59
마이크 - 저성능	11025 Hz	1.65
마이크 - 일반	16000 Hz	3.54
GPS	1 Hz	5.15

## 6.3. 사용자 상황정보와 상태 명세

실제 사용자 상황정보 수집을 위해 4 장에서 설명한 사용자 상황정보 명세 양식과 5 장에서 설명한 사용자 상태 명세 양식을 사용하여 수집하려고 하는 상황정보와 상태를 명세하였다. 명세 된 사용자 상황정보는 7 가지이며, 사용자 상태는 4 가지 이다. 과거 기록 기반의 센서 재구성 기법의 경우, 이전의 기록 데이터가 없다면 작동할 수 없으므로 미리 설정된 값을 넣어주었다. 명세된 사용자 상황정보와 상태는 표 6과 표 7에 서술되어 있다.



표 6 명세된 사용자 상황정보

이름	센서	분석기
Stay	Accelerometer	Standard Deviation
Walk	Accelerometer	Standard Deviation
Run	Accelerometer	Standard Deviation
Salience	Microphone	Decibel
Speech	Microphone	Decibel, Silence Ratio
Music	Microphone	Decibel, Silence Ratio
Meeting	Accelerometer GPS	Standard Deviation Range Matching
Location	Accelerometer Microphone	Standard Deviation Decibel, Silence Ratio

표 7 명세된 사용자 상태

이름	조건 상황정보	관련 센서	재구성 인자
Default	None		
Working	Office, Meeting Office, Stay	가속도계	낮음
Rest	Home, Stay Home, Silence	가속도계 마이크	낮음 낮음
Moving	Walking Running	마이크	낮음

#### 6.4. 실험 시나리오

제안된 모델을 주기적으로 센서를 작동시키는 방법과 비교 하기 위해, 우리는 출근, 업무, 퇴근, 휴식의 4 가지 행동을 반복적으로 수행하도록 실험을 설계하였다. 보다 정확한 실험을 위하여 각 행동들을 취할 때 보여지는 센서의 값들을 미리 추출하였고, 그 센서 값을 실제 센서 값과 바꿔서 제안하는 방법을 특정시간 동안 정확하게 반복 하도록 설계하였다. 출근, 업무, 퇴근, 휴식은 각각 30 분, 120 분, 30 분, 120 분을 주기로 반복된다. 실험 시나리오는 5 번 반복 해서 수행되었다.

#### 6.5. 센서 활성화 주기와 샘플링 레이트

실험 시나리오를 반복하여 수행하는 동안 각 센서의 활성화 주기와 센서의 샘플링 레이트를 측정하였다. 각 센서의 측정결과는 표 8 에 정리되어 있다. 주기적으로 센서를 작동시키는 방법에 비해, GPS 는 스케줄링에 의해 제외되어 적은 작동 횟수를 보여주었으며 모든 센서에서 낮은 평균 샘플링 레이트를 가지는 것을 확인 하였다.

표 8 측정된 센서 활성화 횟수와 샘플링 레이트

	가속도계	마이크	GPS
활성횟수 (주기적)	1044	1044	1044
활성횟수 (제안된 방법)	1043	1043	209
평균 샘플링 레이트 (주기적)	16 Hz	16000 Hz	1 Hz
평균 샘플링 레이트 (제안된 방법)	7.2 Hz	15007.9 Hz	1 Hz

#### 6.6. 소모된 에너지 비교

안드로이드에서 제공하는 Battery API 를 이용하여 주기적으로 센서를 작동 시키는 방법과 제안된 방법을 실험 시나리오를 5 회 반복하는 동안 측정하였다. 비교한 방법은 상황정보를 수집하지 않는 경우, 주기적으로 센서를 활성화 시킨 경우, 제안된 방법이다. 측정 결과는 그림 5 에 도식화 되어 있다. 측정 결과 제안된 방법은 주기적으로 센서를 작동시킨 경우와 비교하여 42% 적은 에너지 소모를 보여주었다.

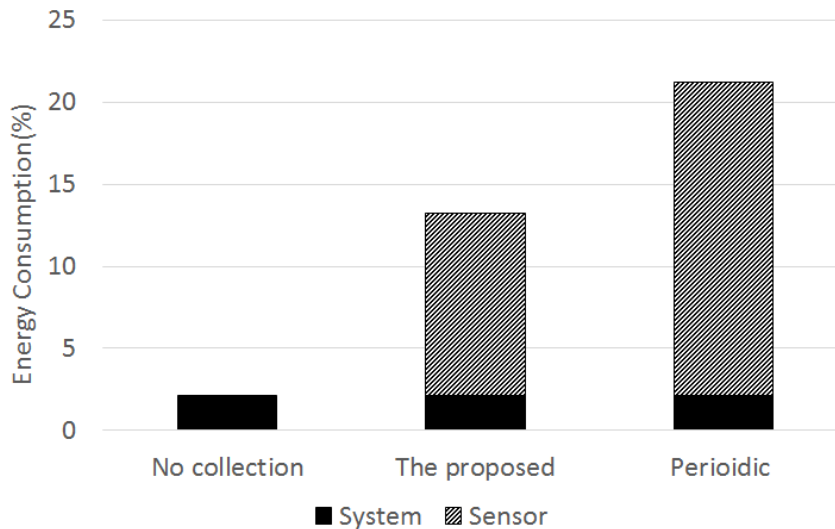


그림 5 각 방법별 에너지 소모 비교

## 6. 결론 및 향후 계획

본 논문에서는 스마트폰을 이용한 사용자 상황정보를 수집하기 위한 방법을 제안하였으며, 제안된 방법은 상황정보 수집에 소모되는 에너지를 줄이는 것을 목표로 하였다. 제안된 방법은 크게 센서 작동 스케줄링 기법과 동적 센서 재구성 기법으로 이루어져 있다. 수집하려고 하는 사용자 상황정보가 명세를 기반으로 센서 활성화 순서를 결정하여 불필요한 센서의 작동을 제외하는 것이 센서 스케줄링 기법의 목표이다. 또한, 동적 센서 재구성 기법의 경우 연속성과 패턴성 이라는 두 가지 단순한 관찰을 응용하여 상황정보 수집에 소모되는 에너지를 줄일 수 있었다. 제안된 방법은 실험을 통하여, 주기적으로 센서를 작동시키는 방법에 비하여 42% 적은 에너지 소모를 보여줌을 확인하였다.

향후 계획으로 제안된 방법을 일반 사용자에게 배포하여 실제로 상황정보를 수집하는 것을 목표로 하고 있다. 일반 사용자에게 배포함으로써 제안된 방법이 실제 사용 환경에서도 효용성이 있다는 것을 검증하는 것이 목표이다. 또한, 제안된 방법을 사용 시, 주기적으로 상황정보를 수집하는 것과 비교하여 수집의 정확도에 큰 영향을 미치지 않는다는 것을 검증 할 필요가 있다. 더불어, 각 센서의 설정 인자를 구성함에 있어 보다 정교하며 다양한 모델을 적용하여 센서에 영향을 주는 보다 다양한 인자들의 최적화를 계획하고 있다. 마지막으로, 현재 제안된 방법은 사용자 상황정보 및 상태를 개발자가 직접 명세해 주어야 하는데, 이를 자동화 하는 방법에 대한 연구를 계속 진행할 계획이다.

## 7. 참고 문헌

- [1] P. TalebiFard and V. Leung, "A dynamic context-aware access network selection for handover in heterogeneous network environments," in Computer Communications Workshops (INFOCOM WKSHPs), 2011 IEEE Conference on, 2011, pp. 385- 390.
- [2] B. Fuentes, A. Bantouna, L. Bennacer, G. Calochira, M. Ghader, G. Katsikas, and F. Yousaf, "On accomplishing context awareness for autonomic network management," in Future Network Mobile Summit (FutureNetw), 2012, 2012, pp. 1- 9.
- [3] A. K. Dey, "Understanding and using context," *Personal Ubiquitous Comput.*, vol. 5, no. 1, pp. 4- 7, Jan. 2001.
- [4] N. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. Campbell, "A survey of mobile phone sensing," *Communications Magazine, IEEE*, vol. 48, no. 9, pp. 140 - 150, sept. 2010.
- [5] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman, "Activity recognition from accelerometer data," in Proceedings of the 17th conference on Innovative applications of artificial

- intelligence - Volume 3, ser. IAAI' 05. AAAI Press, 2005, pp. 1541- 1546.
- [6] Z. Yan, V. Subbaraju, D. Chakraborty, A. Misra, and K. Aberer, "Energyefficient continuous activity recognition on mobile phones: An activityadaptive approach," in ISWC' 12, 2012, pp. 17- 24.
- [7] Y.-S. Lee and S.-B. Cho, "Activity recognition using hierarchical hidden markov models on a smartphone with 3d accelerometer," in Proceedings of the 6th international conference on Hybrid artificial intelligent systems - Volume Part I, ser. HAIS' 11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 460- 467.
- [8] S. Chu, S. Narayanan, and C.-C. J. Kuo, "Environmental sound recognition with time-frequency audio features," *Trans. Audio, Speech and Lang. Proc.*, vol. 17, no. 6, pp. 1142- 1158, Aug. 2009.
- [9] S. Chu, S. Narayanan, and C.-C. Jay Kuo, "Environmental sound recognition using mp-based features," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, 31 2008-april 4 2008, pp. 1 - 4.
- [10] J. Paek, J. Kim, and R. Govindan, "Energy-efficient rate-adaptive gpsbased positioning for smartphones," in *Proceedings of the 8th international conference on Mobile systems, applications, and services*, ser. *MobiSys ' 10*. New York, NY, USA: ACM, 2010, pp. 299- 314.
- [11] Y. Wang, J. Lin, M. Annavaram, Q. A. Jacobson, J. Hong, B. Krishnamachari, and N. Sadeh, "A framework of energy efficient mobile sensing for automatic user state recognition," in *Proceedings of the 7th international conference on Mobile systems, applications, and services*, ser. *MobiSys ' 09*. New York, NY, USA: ACM, 2009, pp. 179- 192.
- [12] H. Lu, J. Yang, Z. Liu, N. D. Lane, T. Choudhury, and A. T. Campbell, "The jigsaw continuous sensing engine for mobile phone applications," in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, ser. *SenSys ' 10*. New York, NY, USA: ACM, 2010, pp. 71- 84.
- [13] Y. Lee, Y. Ju, C. Min, S. Kang, I. Hwang, and J. Song, "Comon:cooperative ambience monitoring platform with continuity and benefit awareness," in *Proceedings of the 10th international conference on Mobile systems, applications, and services*, ser. *MobiSys ' 12*. New York, NY, USA: ACM, 2012, pp. 43- 56.



**한 윤 선**

2009 포항공과대학교, 컴퓨터공학과 학사

2009 ~현재 포항공과대학교, 정보전자융합공학부 석/박사 통합 과정

<관심분야> Software Defined Networking (SDN), 상황인지 서비스, 정보 모델링



**강 준 명**

2005 포항공과대학교 컴퓨터공학과 학사

2011 포항공과대학교 컴퓨터공학과 박사

2011 포항공과대학교 컴퓨터공학과 박사후연구원

2011 ~현재 토론토대학교 전자컴퓨터공학과 박사후연구원

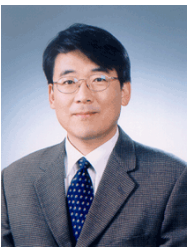


**서 신 석**

2008 인하대학교, 컴퓨터공학과 학사

2008~현재 포항공과대학교, 컴퓨터공학과 통합 과정

<관심분야> Software Defined Networking (SDN), 데이터 센터 네트워킹, 상황정보 관리



**홍 원 기**

1983 Univ. of Western Ontario, BSc in Computer Science

1985 Univ. of Western Ontario, MS in Computer Science

1985 ~ 1986 Univ. of Western Ontario, Lecturer

1986 ~ 1991 Univ. of Waterloo, PhD in Computer Science

1991 ~ 1992 Univ. of Waterloo, Post-Doc Fellow

1992 ~ 1995 Univ. of Western Ontario, 연구교수

1995 ~현재 포항공과대학교 컴퓨터공학과 교수

2007~2011 포항공과대학교 정보통신대학원장

2007~2010 포항공과대학교 정보통신연구소 연구소장

2008~2010 포항공과대학교 컴퓨터공학과 주임교수

2008~2012 포항공과대학교 정보전자융합공학부장

2008 ~현재 포항공과대학교 정보전자융합공학부 교수

2012 ~현재 KT 종합기술원 원장