

## 웨어러블 컴퓨터를 위한 Self-Configuration Manager

박충범<sup>o</sup>, 이수원, 최 훈  
충남대학교 컴퓨터 공학과  
{here4you<sup>o</sup>, swlee, hc}@cnu.ac.kr

### Self-Configuration Manager for Wearable Computers

Choong-Bum Park<sup>o</sup>, Su-Won Lee, Hoon Choi  
Dept. of Computer Engineering, Chungnam National University

#### 요 약

I/T 산업의 발전으로 다양한 종류의 디바이스와 시스템 소프트웨어 및 응용프로그램이 일반화되면서 시스템의 복잡도가 높아지는 문제점이 발생하였고 이러한 문제는 시스템 운영관리를 위한 전문지식이 없는 일반 사용자에게는 더욱 치명적이다. 이러한 문제점을 해결하기 위해서 Autonomic Computing 개념이 연구되고 있다. 본 연구에서는 보다 개인화되고 자원제약적인 웨어러블 컴퓨팅 환경에 적합한 Autonomic Personal Computing을 위한 자율관리 서비스 구조를 제안한다. 본 논문에서 제안하는 Configuration Manager는 Self-Configuration을 위한 필수 기능만을 Client에 탑재하고, 많은 리소스 소비를 요구하는 기능은 Server의 도움을 받는 구조로 자원제약적인 웨어러블 컴퓨팅 환경에서 원활한 Self-Configuration을 제공한다.

#### 1. 서 론

I/T산업의 눈부신 발전에 의해 컴퓨터 시스템은 보다 작고 빠르고 저렴해졌으며, 다양한 종류의 디바이스와 시스템 소프트웨어 및 응용프로그램을 이용하여 고도의 정보 처리가 가능해졌다. 하지만 기능 향상과 함께 시스템의 복잡도(complexity)도 높아지는 문제점이 발생하였다. 데스크톱의 일반화, 웨어러블 컴퓨터의 등장으로 일반사용자의 정보화는 신속하게 진행되고 있지만 전문지식이 없는 일반 사용자들에게 시스템 운영관리라는 부담이 발생하였다. 특히 웨어러블 컴퓨팅 환경의 경우 착용한 상태에서 활동하는 사용자가 복잡한 관리절차를 차분하게 수행하기를 기대하기 어렵다. 이렇게 다양한 형태의 컴퓨팅 환경과 디바이스들의 출현에 의해 더 이상 사람이 직접 시스템을 관리할 수 없는 상황이 되었다. 이에 따라 IBM에서는 시스템 스스로 자신을 관리하는 Autonomic Computing이라는 개념을 발표하였다[1]. 하지만 IBM의 Autonomic Computing은 서버급 이상의 컴퓨터를 대상으로 하므로 개인화된 컴퓨팅 환경에는 적합하지 않으며 퍼스널 컴퓨팅 시스템에 Autonomic Computing 개념을 도입한

Autonomic Personal Computing에 대한 연구가 진행되고 있다[2][3].

본 논문은 보다 개인화되고 자원제약적인 소형 모바일 디바이스를 이용한 웨어러블 컴퓨팅 환경에 적합한 Autonomic Personal Computing을 위한 자율적 관리(Self-Configuration) 서비스 구조를 제안한다.

본 연구에서 제안하는 Configuration Manager는 웨어러블 컴퓨터를 대상으로 하므로 Self-Configuration을 위한 필수적인 기능만을 Client에 탑재하고, 많은 리소스와 컴퓨팅 파워를 필요로 하는 기능은 Home Server의 도움을 받는다.

2장에서는 관련연구로서 기존의 Autonomic Computing 시스템의 특징을 기술하고, 3장에서는 본 연구의 선행연구인 웨어러블 미들웨어 프레임워크(WMF: Wearable Middleware Framework)에 대해서 기술한다. 4장에서는 WMF에 적합한 Self-Configuration을 위해 제안하는 Configuration Manager에 대해서 기술하고 5장에서는 결론과 향후 과제에 대해서 기술한다.

본 연구는 정보통신부의 선도기반기술개발사업의 지원으로 수행되었습니다.

2. 관련 연구

I/T산업의 발전에 따른 생산성 향상과 함께 복잡도가 증가되어 더 이상 관리자가 시스템을 효율적으로 관리할 수 없는 상황이 초래하였고 시스템 스스로가 지능적으로 자신을 관리하는 Autonomic Computing 개념이 생겼다. Autonomic Computing의 특징은 다음과 같다.

[표 1] Autonomic Computing 특징

특징	설명
Self-Configuring	시스템 스스로 시스템을 재구성
Self-Healing	시스템 스스로 시스템 장애 복구
Self-Optimizing	시스템 스스로 시스템의 최적화
Self-Protecting	시스템 스스로 시스템을 보호

IBM에서는 Autonomic Computing을 수행하는 Autonomic Manager(AM)를 구현하였으며 다음과 같은 구성을 가진다.

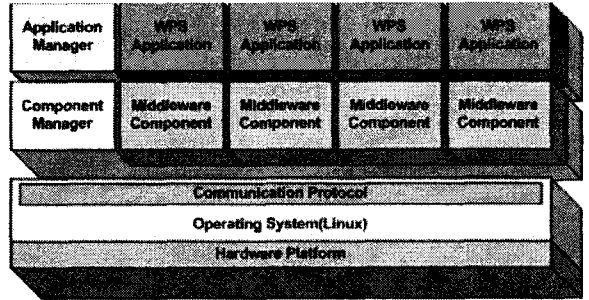
[표 2] Autonomic Manager의 기능구성

구성	기능
Monitor	관리대상으로부터 Log를 수집하고 필터링하여 Analyze Part로 보고
Analyze	보고받은 정보를 모델화하고 연관시키는 분석작업 수행
Plan	목표 이행을 위해 필요한 행동을 생성
Execute	Plan Part에서 만들어진 행동을 실행

IBM의 AM은 서버급 컴퓨터를 대상으로 하였으나 보다 전문적으로 관리되기 어려운 PC환경에 AM기술을 응용한 Autonomic Personal Computing에 대한 많은 연구가 진행되고 있으며 보다 개인화되고 소형화되며 이동성을 가지는 웨어러블 디바이스환경에 적합한 Autonomic Computing에 대한 연구도 진행되고 있다[5].

3. WMF(Wearable Middleware Framework)

본 연구팀은 한국전자통신연구원과의 공동연구로 웨어러블 컴퓨팅환경에 적합한 미들웨어인 WMF를 개발하였으며 한국정보통신기술협회(TTA)를 통해 표준화를 진행하고 있다[5]. WMF 프레임워크의 구조는 [그림 1]과 같다.



[그림 1] WMF 프레임워크

WMF는 다음과 같은 특징을 가진다.

- 웨어러블 컴퓨팅환경에 적합한 미들웨어
 

WMF는 저전력, 자원제약적인 환경을 고려하여 구현되었다. 모든 컴포넌트는 C언어로 개발되었으며 사용자의 목적에 맞는 컴포넌트와 WPS(Wearable Personal Station, 웨어러블 컴퓨터) 어플리케이션만으로 프레임워크를 구성하여 불필요한 리소스 낭비를 방지할 수 있다.
- 컴포넌트 기반의 구조
 

WMF는 구성하는 모든 미들웨어컴포넌트 및 어플리케이션은 컴포넌트형태로 개발되어 런타임 중에도 프레임워크의 재컴파일 없이 확장이 가능하다[6].
- 미들웨어컴포넌트 및 어플리케이션의 명세
 

모든 컴포넌트와 어플리케이션은 자신의 정보(이름, 제조사, 버전, 어플리케이션 및 컴포넌트간 종속성)를 나타내는 CDF(Component Description File)과 함께 배포된다.
- 동적 재구성
 

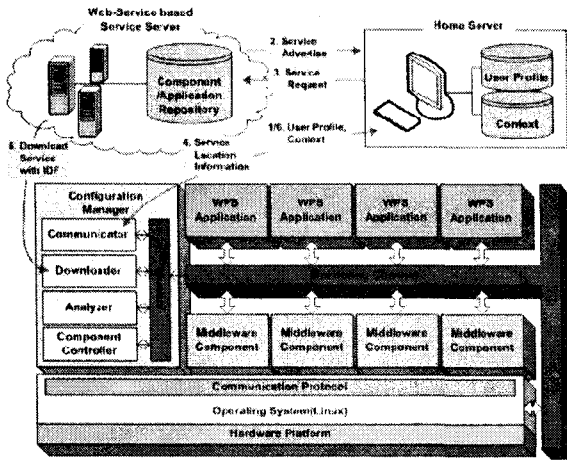
WMF의 동작 중에 필요로 하는 어플리케이션 및 미들웨어컴포넌트를 검색하고 설치/실행/종료/갱신/삭제 가능하다.

WMF는 웨어러블 컴퓨팅환경을 고려하여 사용자의 목적에 따라 개인화된 프레임워크의 재구성이 가능하다는 장점이 있으나 재구성에 필요한 모든 의사결정에 사용자가 직접 의사를 반영해야 한다는 부담이 있다. 이를 해결하기 위해 프레임워크의 재구성에 사용자의 개입을 최소화할 수 있는 Configuration Manager를 설계하였다.

4. WMF를 위한 Self-Configuration Manager

4.1 Self-Configuration Manager

Self-Configuration Manager의 구조는 [그림 2]와 같다.



[그림 2] Self-Configuration Manager

Communicator, Downloader, Analyzer는 WMF의 Component Manager에 존재하던 모듈로서 사용자의 개입을 통하여 프레임워크의 재구성을 수행하였다. 이 모듈들은 Component Manager에서 분리되어 Self-Configuration을 수행하는 핵심 모듈로 확장되었다. 각 모듈의 기능은 다음과 같다.

- **Communicator:** 주기적인 통신을 통해 WMF의 User Profile과 Context를 서버와 주고받는다.
- **Downloader:** 서버로부터 어플리케이션과 미들웨어컴포넌트를 다운로드하는 모듈이다. 이때 어플리케이션/미들웨어컴포넌트의 정보를 명세한 CDF(Component Description File)가 함께 다운로드된다.
- **Analyzer**
  - Monitor로부터 제공된 상태정보를 상태분석규칙(State Analysis Rule)을 기반으로 분석하여 Context를 생성하여 Home Server에게 제공한다.
  - CDF파일을 분석하여 프레임워크 재구성을 위한 정보를 Component Controller에게 제공한다.
- **Monitor:** 새롭게 추가된 컴포넌트로서 프레임워크 및

시스템의 정보를 수집하는 Monitoring Channel을 생성하고 관리한다.

- **Component Controller:** 기존 WMF의 Application Manager와 Component Manager를 통합한 모듈로서 어플리케이션과 미들웨어컴포넌트를 설치/실행/중료/갱신/삭제하는 컴포넌트이다.
- **WPS Application:** WMF에서 동작하는 어플리케이션
- **Middleware Component:** WMS 어플리케이션을 지원하는 미들웨어가 구현된 컴포넌트

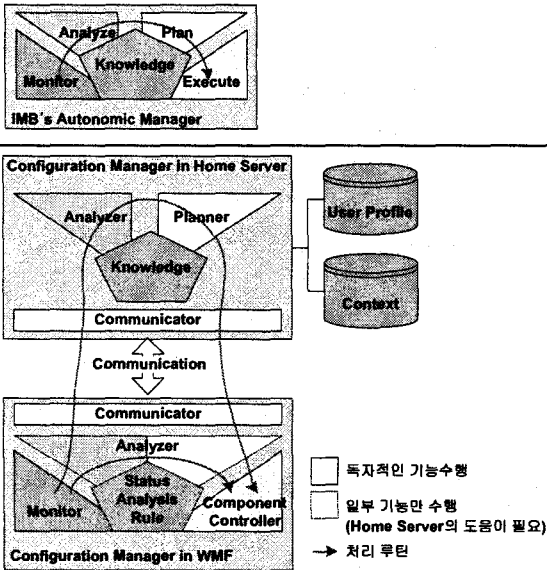
Self-Configuration을 위해 필요한 monitoring, anaysis, planning, execution 기능을, 소형이며 메모리나 프로세서 성능이 제한적인 웨어러블 컴퓨터에서 모두 수행시키는 것은 웨어러블 컴퓨터의 고유한 기능 수행에 큰 오버헤드가 될 수 있다. 따라서 이들 중 일부 기능을 웨어러블 컴퓨터를 대신하여 수행하는 네트워크 내 서버 시스템이 필요하다.

- **Home Server:** Configuration Manager와 지속적으로 대화하며 User Profile과 Context를 저장하며 Configuration Manager의 Self-Configuration을 지원하는 시스템
- **Service Server:** WMF에 설치 가능한 미들웨어컴포넌트/어플리케이션과 그 명세인 CDF를 제공하는 시스템

Configuration Manager를 구성하는 모듈들을 IBM의 Autonomic Manager의 구성과 비교하면 <표 3>과 같고 처리과정을 비교하면 [그림 3]과 같다.

<표 3> WMF Configuration Manager와 IBM Autonomic Manager 기능 비교

Autonomic Manager 기능	해당 Configuration Manager 기능
Monitoring	Monitor
Analysis	Analyzer & Home Server
Planning	Home Server
Execution	Component Controller



[그림 3] WPS Configuration Manager과 IBM Autonomic Manager 처리과정 비교

## 4.2 동작 시나리오

### 4.2.1 Monitoring/Analysis/Communication

WMF의 구동과 함께 Monitor는 Monitoring Channel을 통해 프레임워크와 시스템의 상태를 감시한다[6].

- 프레임워크의 감시

프레임워크를 구성하는 모든 미들웨어컴포넌트/어플리케이션의 감시용 Interface를 주기적으로 invocation 하고 그에 대한 응답을 받는 방식을 이용한다. Monitor의 invocation에 응답하지 않는 active한 상태의 미들웨어컴포넌트/어플리케이션은 장애가 발생된 것으로 간주하고 Monitor에 감지되어 Analyzer로 보고된다.

- 시스템 감시

미들웨어컴포넌트/어플리케이션의 프로세스 상태 정보 및 하드웨어 상태를 수집하는 방식을 이용한다. 특정 미들웨어컴포넌트/어플리케이션의 프로세스가 정상적으로 동작하지 않거나 컴퓨팅환경의 상태변화(네트워크 연결상태-WLAM/BAN, 디바이스 연결상태-새로운 디바이스의 플러그/언플러그)가 발생하면 그 정보가 Monitor를 통하여 Analyzer로 통보된다.

Analyzer는 Monitor로부터 제공받은 상태정보를 상태분석규칙을 기반으로 분석하여 User Profile과 Context를 생성하여 Communicator를 통해 Home Server로 전송한다.

- User Profile: 사용자와 사용자가 사용하는 서비스목록과 같은 비교적 정적인 정보
- Context: 실행환경이나 사용중인 서비스의 상태와 같은 비교적 유동적인 정보

Analyzer의 분석결과 신속한 재구성을 필요로 하는 경우 Component Controller에게 통보되어 해당 미들웨어컴포넌트/어플리케이션을 종료하거나 재실행하여 위급한 상황에 대처한다[7]. 그 외의 경우에는 Communicator를 통해 Home Server로 생성한 User Profile과 Context를 전송한다. 필요에 따라 Home Server에 저장된 User Profile과 Context를 받아들 수 있다.

### 4.2.2 Self-Reconfiguration

Configuration Manager는 Communicator를 통해 Home Server와 User Profile과 Context를 지속적으로 주고 받는다[그림 2에서의 단계1]. Service Server는 자신이 가지고 있는 서비스(미들웨어컴포넌트/어플리케이션)를 Home Server에게 광고하고[그림 2\_2] Home Server는 WMF를 대신하여 User Profile과 Context를 기반으로 WMF에 적절한 서비스를 선택하여 Service Server에게 서비스를 요청한 후[그림 2\_3] 요청한 서비스의 정보를 WMF의 Communicator에게 전달한다[그림 2\_4]. Downloader는 Communicator로부터 서비스 정보를 넘겨 받아 Service Server에 접속하여 서비스와 CDF를 전송 받게 된다[그림 2\_5]. Analyzer는 전송 받은 CDF를 분석하여 서비스 설치 이전에 선행되어야 할 작업과 설치방법을 결정하여 Component Controller에게 전달한다. Component Controller는 Analyzer로부터 전달받은 정보를 기반으로 실제 미들웨어컴포넌트/어플리케이션을 프레임워크에 설치하고 실행한다. 미들웨어컴포넌트/어플리케이션의 설치 및 실행결과는 User Profile과 Context로 작성되어 Communicator를 통해 Home Server로 전송되고 [그림 2\_6] Home Server의 저장소에 저장된다.

#### 4.2.3 Self-Healing

Configuration-Manager는 WMF 동작 도중 발생하는 장애에 대처한다. Monitor를 통해 Analyzer로 제공된 상태 정보가 장애로 판단되게 되면 Component Controller에게 통보된다. Component Controller는 장애가 발생한 미들웨어컴포넌트/어플리케이션과 다른 미들웨어컴포넌트/어플리케이션과의 종속관계를 판단하여 재구성의 범위를 결정 한 후에 해당 미들웨어컴포넌트/어플리케이션들을 재실행한다. 같은 컴포넌트/어플리케이션에서 지속적으로 장애가 발생하면 Home Server와 상의하여 재설치하거나 갱신한다.

#### 4.2.4. Self-Optimization

WMF의 컴퓨팅 환경변화에 따른 재구성도 필요하다. 가용메모리의 크기나 배터리 잔여용량이 임계값 이하로 줄어들면 실행중인 어플리케이션 중에 낮은 우선순위를 가지는 어플리케이션이 종료되고 필요한 경우 메모리에서 제거된다. 네트워크 연결상태에 변화가 발생하면 네트워크 기능을 이용하는 어플리케이션이 종료된다. WMF의 컴퓨팅 환경이 개선되게 되면 Component Controller에 의해 종료되었던 어플리케이션이 재시작된다. 어플리케이션의 우선순위나 네트워크 사용여부 등에 대한 정보는 CDF에 기술된 내용을 기준으로 한다.

#### 5. 결론 및 향후 과제

본 연구에서는 웨어러블 컴퓨팅 환경에 적합한 WMF를 개발하였다. 본 프레임워크는 런타임 시에 미들웨어컴포넌트와 어플리케이션을 설치/실행/종료/갱신/삭제 할 수 있는 장점을 지니지만 모든 재구성 과정에 사용자의 의사결정이 필요하다는 단점이 있다. 이러한 WMF에 Autonomic Computing개념을 도입하여 프레임워크의 재구성에 사용자의 개입을 최소화하여 Self-Configuration을 제공하기 위한 Configuration Manager를 제안하였다.

WMF는 소형 모바일 디바이스를 대상으로 하기 때문에 IBM의 Autonomic Manger에 대응하는 모든 기능을 수용하기는 불가능하므로 필수적인 기능은 WMF의 Configuration Manager에서 수행하고 그 외의 기능은 Home Server가 대행하는 구조를 가진다. 이에 따라 자원 및 컴퓨팅파워의 제약을 받는 시스템에서도 원활한 Self-

Configuration이 가능하다.

본 연구의 향후 과제는 다음과 같다.

- User Profile/Context, CDF의 표준포맷 정의 및 적용
- Home Server/Service Server와 WMF간의 보안통신
- 피드백을 통한 강화학습 알고리즘
- 서비스 Vendor와의 정보교환 Interface

#### 6. 참고문헌

- [1] Horn, Paul, " Autonomic Computing: IBM' s Perspective on the State of Information Technology" , available form the IBM Corporation at <http://www.research.ibm.com/autonomic/manifesto/autonomiccomputing.pdf>
- [2] Bantz D, Frank D, " Challenges in autonomic personal computing with some new results in automatic configuration management" , Industrial Informatics 2003, pp. 451-456, Aug.2003
- [3] Bantz D, Frank D, " Autonomic personal computing" , available form the IBM Corporation at <http://researchweb.watson.ibm.com/journal/sj/421/bantz.html>
- [4] Sterritt R, Bantz D, " Personal autonomic computing reflex reactions and self-healing" , IEEE Transactions on Systems, Man and Cybernetics, Part C, VOL.36, No3, pp. 304-314, May.2006
- [5] 유용덕, 최 훈, 김형신, 권영미, Takeshi Nanri, "경량 미들웨어를 위한 소프트웨어 구조", 한국차세대PC 논문지 제1권 제2호, pp. 34-42, 2005.12.
- [6] 이수원, 유인선, 유용덕, 최 훈, " 컴포넌트 기반 소프트웨어 플랫폼을 위한 컴포넌트 관리자 설계 및 구현", 한국정보과학회 KCC2006, 논문집(D), 제33권 제1호, pp. 199-201, 2006. 6.
- [7] 윤현상, 박정민, 이은석, " 유비쿼터스 환경에서의 상황인식형 자가치유 시스템" , 한국지능정보시스템학회 2004 추계학술대회, pp.259-268, Nov.2004