

Context-Aware 어플리케이션을 위한 효율적인 위젯 관리

황정섭⁰ 류은석 유혁
고려대학교 운영체제 연구실
{jshwang⁰, esryu, hxy}@os.korea.ac.kr

An Efficient Management for Context-Aware Applications

Jeongseop Hwang⁰ Eunseok Ryu Chuck Yoo
Operating System Laboratory, Korea University

요 약

최근 들어 유비쿼터스 관련 연구 개발을 통해 문맥 정보를 받아들이고 이를 해석하고 분석한 후, 상황에 맞는 적절한 대응을 해주는 미들웨어의 연구가 많이 이루어지고 있다. 하지만, 미들웨어 입장에서 센서를 감싸는 위젯(widget)을 효과적으로 관리하는 연구가 부족하다. 이런 이유로, 우리는 이러한 위젯의 효과적인 유지 관리를 통해 미들웨어에서 가질 수 있는 장점을 알아보고 이를 지원하기 위한 위젯 통합 프레임워크를 제안 하였다.

위젯 통합 프레임워크는 다양한 위젯을 통합하고 문맥 정보를 관리하기 위하여 위젯의 Runtime Binding, Error Resilience, Common Interface over Widget을 지원한다. 또한 서비스를 관리하는 서비스 매니저와 위젯 사이의 상호작용을 효율적으로 지원함으로써 상위 계층으로 보다 적당한 문맥 정보를 제공할 수 있다. 그리고 위젯 기반의 프레임워크를 구성함으로써 다른 미들웨어에서 제공하지 못한 위젯 중심의 기능들을 제공한다.

1. 서 론

유비쿼터스 컴퓨팅 환경은 사용자들로 하여금 컴퓨터나 기기기간의 장치들로 인한 혼란을 겪지 않도록 해준다. 이러한 상황에서 사용자의 주변 환경, 이를테면, 온도, 밝기, 습도와 같은 환경 요소들은 사용자가 다른 행동 패턴을 가지도록 한다. 따라서 이러한 환경 정보는 사용자의 패턴을 파악하고 적절한 서비스를 제공하는데 있어서 중요한 정보가 된다. 때문에 이러한 환경 정보를 얻을 수 있는 Environment Sensing의 중요성이 더욱 커진다.

주변의 다양한 환경 정보를 얻기 위해서 각 어플리케이션이 센싱 장치 하나하나에 직접 접근하고 정보를 얻어 오는 것은 불합리한 점이 많다. 개발자의 입장에서 하나의 어플리케이션을 개발하기 위해 어플리케이션 본래의 기능을 구현하는 것에 충실하지 못하고 센싱 장치에 접근하고 정보를 얻어 오는 작업에 많은 시간을 투자해야 한다. 이것은 결국 개발자로 하여금 시간과 인력의 낭비를 초래하게 한다. 따라서 환경 정보를 얻을 때에, 각 센싱 장치에 접근하여 정보를 얻어오는 미들웨어(middleware) 계층이 존재하고, 이들의 상위 어플리케이션이나 혹은 다른 미들

웨어에게 필요한 정보를 제공해 줄 수 있다면, 어플리케이션을 개발하는 데 있어서 많은 수고를 덜 수 있다.

우리는 이러한 위젯의 연구를 통해 기존의 미들웨어에서는 얻지 못하는 장점을 알아보고 이를 지원하는 위젯 통합 프레임워크 (Widget Integration Framework)를 제안한다. 위젯 통합 프레임워크는 다른 미들웨어에서는 제공되지 않는 다음과 같은 특징이 있다.

첫째, 미들웨어에서 서비스 검색 도구가 서비스에 맞는 위젯을 선택하여 이를 통해 필요한 데이터를 전해 받을 수 있는 기능을 두고자 한다. 이를 위해서는 현재 가용할 수 있는 위젯들을 파악하고 이를 전달해줄 수 있는 방안을 마련한다. 이를 이용하면 큰 범위의 동적 재구성을 지원하는 발판이 될 수 있다.

둘째, 현재의 미들웨어 시스템에는 하드웨어로 구성된 다량의 센서 중에서 발생할 수 있는 오류로 인해 전체 시스템에 장애를 가져올 수 있다. 따라서 적어도 위젯 관리 차원에서 이러한 오류 발생 위젯을 찾아서 이를 서비스에 연결되지 않도록 조치하는 일이 필요하며 이를 통해 고장에 강한(fault-tolerant) 미들웨어를 지원할 수 있다.

이상의 기능을 제공함으로써 상위 미들웨어 계층으로

보다 신뢰성 있고 적절한 데이터, 효율적인 문맥 정보의 관리를 지원할 수가 있다.

2. Related Work

2.1 Context Toolkit

Context Toolkit[1]은 유비쿼터스 환경 하에서 주위에 존재하는 컨텍스트 - 하나의 엔티티의 상황을 특징짓는데 사용될 수 있는 임의의 정보 - 를 쉽게 이용하기 위해 상위 어플리케이션에게 하위 컨텍스트 정보를 제공하기 위한 미들웨어이며, Georgia Institute of Technology의 Day에 의해 진행되었다. Context Toolkit은 하위 컨텍스트들에 대한 적절한 추상화를 통해서 컨텍스트가 로컬에 있던 리모트에 있던지 상관없이 어플리케이션이 이용할 수 있는 방법을 제공한다.

Context Toolkit은 크게 세 가지 부분, 위젯, Aggregator, 인터프리터로 구성되어 있다.

위젯은 컨텍스트를 사용하는 어플리케이션이 그 하단의 세부적인 내용을 알 필요가 없도록 단일화된 인터페이스로써 컨텍스트를 이용할 수 있도록 하기 위한 개체이다. 보통 하나의 컨텍스트를 래핑하고 있다. Aggregator는 기본적인 위젯들과 응용 프로그램 사이의 통로(gateway) 역할을 함으로써 어플리케이션이 두 개 이상의 여러 위젯의 정보를 통합(integrate) 할 수 있도록 한다. 그리고 인터프리터는 하위 개념의 컨텍스트 정보를 보다 상위 개념의 정보로 해석해 주는 개체이다.

3. Widget Integration Framework

3.1 WIF Architecture

주변 환경을 센싱하여 생성된 문맥 정보를 받아들여 이를 해석하고 분석하여 상황에 맞는 적절한 대응을 할 수 있도록 미들웨어에 전달하는 역할을 하도록 하기 위해, 센싱 모듈을 감싸는 위젯을 구현하고 그들을 통합하는 위젯 통합 프레임워크(WIF) 구조를 설계하였다.

그림 1에서 보는 바와 같이 WIF는 크게 Widget Information Manager(WIM)와 Registration Manager, Fault-detection Manager, Aggregator, 그리고 각종 위젯 등으로 구성되어 있다. 현재 각종 위젯들을 개발하여 미들

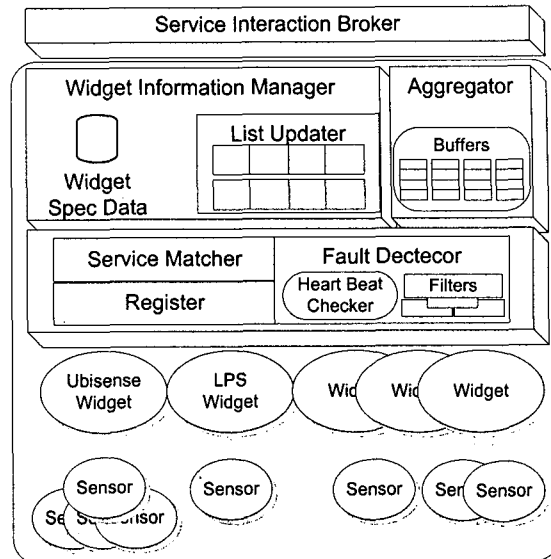


그림 1 The WIF Architecture

웨어의 서비스 상호작용 브로커(SIB)에 직접 연결하는 부분까지 구현하였고, 다른 구성 요소들에 대해서는 기초 조사 및 요구 분석을 통한 결과를 토대로 구현 중이다.

3.2 WIM (Widget Information Manager)

WIM은 여러 위젯들의 현재 상태를 기록 및 관리하는 역할을 한다. 따라서 위젯들이 새로 등록되거나 제거되었을 때, 또한 문제가 생길 때 이러한 정보를 Registration Manager와 Fault-detection Manager를 통해 업데이트 메시지(update message)로 전달 받는다. 이러한 WIM은 다시 Service Manager와 정보를 주고 받으며 응용프로그램이 원하는 서비스를 지원할 수 있는지 여부를 알려준다. 다시 말해서, 특정 서비스에 필요한 적절한 위젯을 서비스 검색에 매칭 시켜주는 역할을 할 수 있다.

3.3 Registration Manager

Registration Manager는 새로운 위젯이 미들웨어에 등록되고자 할 때 미리 정의된 프로토콜에 따라 이를 등록하는 역할을 한다.

여러 센서와 위젯들이 잘 갖추어진 환경에 새로운 위젯이 새로 등록될 필요가 있을 때가 있고, 때때로 제거될 때가 있다. 이와 같이 위젯들이 등록되거나 제거될 때 이를 관리하여 WIM에 새로운 업데이트 메시지를 통해 알려주는 역할을 Registration Manager가 맡는다.

3.4 Fault Detection Manager

Fault-detection Manager는 위젯 또는 센싱 장비에 발생할 수 있는 어떠한 오류 및 문제점에 대해 이를 감지하여 WIM에 알리는 역할을 한다.

안정된 구조로 돌아가는 미들웨어 및 기타 하드웨어 장비일지라도 때로는 예상치 못하는 상황으로 인해 작동이 되지 않을 수 있다. 센서 혹은 펌웨어와 위젯간의 유무선 데이터 통신 중에 오류가 발생할 여지가 있으며, 따라서 WIF에서는 이러한 위젯 하부 계층의 문제 발생에 대해, 동작 중인 센서에 대해서는 받아들이는 데이터를 통해서, 그리고 현재 동작하지 않고 대기 중인 센서에 대해서는 주기적인 체크 메시지를 통해 이러한 센서 및 위젯들이 언제 어디서든 여러 서비스를 제공할 준비가 되어 있음을 WIM으로 알릴 수 있다. 이와 같이, 이러한 오류를 극복할 수 있고 감내할 수 있는 미들웨어를 지원하는 것이 Fault-detection Manager의 목적이다.

3.5 Aggregator

CTK(Context Toolkit) 등에서 사용된 Aggregator는 여러 위젯으로부터 받아들이는 데이터를 효과적으로 관리하여 응용프로그램에 전달하는 역할을 한다. 우리는 이러한 Aggregator의 아이디어를 기반으로 좀 더 확장된 기능을 하는 모듈을 구상하고 있다. 특히, 현재 대부분의 유비쿼터스 미들웨어에서 컨텍스트의 위치 정보 및 주변 온도, 습도 등 데이터 사이즈가 그리 크지 않는 정보만을 취급하는 것에 비해, 앞으로는 여러 감시 카메라나 마이크 장치 및 3D 스캐너 등에 입력되는 엄청난 양의 데이터를 미들웨어 및 응용프로그램에서 적절하게 이용할 일이 많아질 것이다. 즉, 영상 인식 및 음성지문 인식 등의 일들을 미들웨어 및 응용프로그램에서 처리한다면 이러한 다량의 데이터를 효과적으로 분류하여 제거할 것은 제거하고 중복되는 데이터를 골라내며 필요한 데이터를 우선적으로 전달하는 발전된 형태의 Aggregator가 필요할 것으로 내다본다. 따라서 WIF에 이러한 Aggregator를 연구 및 구현하여 추가시킬 예정이다.

3.6 Widget

배경 연구에 언급된 컨텍스트 툴킷에서 일반적으로 사

용되는 위젯의 개념은 컨텍스트를 사용하는 애플리케이션이 그 하단의 세부적인 내용을 알 필요 없이 단일화된 인터페이스로써 컨텍스트를 이용할 수 있도록 하기 위한 개체이다. 따라서 WIF에서의 위젯의 개념 역시 하단의 센서로부터 읽은 데이터를 상위 계층으로 전달하는 역할을 수행함으로써 상위 계층에 대한 하부의 투명성을 제공하는 것이다. WIF에서의 위젯은 이러한 기능에 Registration 매니저 및 Fault-detection 매니저와의 인터랙션 기능이 추가되었다.

4. 결 론

이상의 연구를 통해 우리는 컨텍스트를 래핑하고 있는 Widget에 좀 더 특화된 프레임워크를 제안함으로써 기존의 미들웨어에서 제공해 주지 못했던 서비스들을 지원해 줄 수 있다. 이러한 위젯 통합 프레임워크는 본문에서 언급했듯이 Dynamic Binding, Error Resilience의 기능을 가진다.

이러한 특징을 갖는 WIF는 그 많은 계층의 특성으로 인해 전체 시스템에 큰 오버헤드를 주지 않으면서 서비스를 제공할 수 있다. 또한 이를 확장하여 서로 다른 종류의 위젯을 통합하는 일도 할 수 있을 것이다. 이를 통해 다양한 종류의 미들웨어와 위젯이 사용되는 환경에서 WIF를 사용하여 보다 간단하고 편안하게 위젯을 사용할 수 있다. 이렇게 우리는 이기종간의 위젯들을 통합하는 작업을 최종의 목표로 삼고 있다. 앞으로 펼쳐질 수많은 센서와 컨텍스트가 존재하는 상황에서 하위의 위젯을 어떻게 사용하고 관리할 것인지는 중요한 이슈가 될 수 밖에 없으며, 우리는 이 문제를 해결함으로써 애플리케이션 개발자로 하여금 보다 쉽고 간단히 컨텍스트의 정보를 이용할 수 있도록 하였다.

5. 참고문헌

1. Anind K. Dey, Daniel Salber and Gregory D. Abowd, "A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications", Human-Computer Interaction (HCI) Journal, Volume 16 (2-4), pp. 97-166, 2001.