

유비쿼터스 데이터 관리를 위한 Cyber View 명세화 도구개발

박상현*, 민수영, 고재진, 주현태**
전자부품연구원*, 한국외국어대학교**

An Implementation of Cyber View schema tool for Ubiquitous Data Integration and Management

Sang Hyun Park *, Sooyoung Min, Jae Jin Ko, Hyun Tae Ju**
Korea Electronics Technology Institute*, Hankuk University of Foreign Studies**
E-mail : {*shpark, minsy, jaejini}@keti.re.kr, cisc@hufs.ac.kr**

Abstract

The rapid growth of Hardware technologies and Network fitted to us Ubiquitous Computing Life[1]. Ubiquitous Computing integrates computation into the environment, rather than having computers which are distinct objects. There are many computational things like a Sensor Network, RFID, GPS, Mobile devices, and so on. Ubiquitous Data Integration and Management are new paradigms. The goals of UDI Service Platform are data protection for the distributed data on pervasive computing devices and data distribution to appropriate users with best distribution policies. To implement the idea we evaluate the logical schema Cyber View that is a management tool.

I. 서론

네트워크 및 하드웨어 기술의 혁신적인 발전으로 유비쿼터스 컴퓨팅 환경 실현이 가시화되고 있다. 유비쿼터스 컴퓨팅 환경은 유무선의 다양한 네트워크와 컴퓨터, 휴대전화, 센서네트워크, RFID, GPS 에 이르는 다양한 디바이스들을 실제공간상에 사용하여 효과적인 사이버 공간을 제공하는 사용자 중심의 컴퓨팅 환경을 의미한다. 이와 같은 유비쿼터스 컴퓨팅 환경에서 데이터를 효율적으로 관리하는 것이 새로운 패러다임으로 떠오르고 있다.

본 논문에서는 개개인의 위치/컨텍스트에 따른 효과적인 데이터 분배 및 보호 등을 모두 제공하기 위해 UDI (Ubiquitous Data Integration) 서비스 플랫폼의 개발을

소개하고, 사용자의 데이터, 응용 및 이들을 수행하는 컴퓨팅 자원의 관리 정책인 논리적 스키마 Cyber View 정의를 위한 명세화 도구 개발에 대해 기술하였다.

II. 본론

UDI 서비스 플랫폼[그림 1]은 유비쿼터스 컴퓨팅 환경에서 언제 어디서, 어떠한 디바이스 환경에서든 동일한 데이터 서비스 환경을 제공하고, 사용자의 위치/컨텍스트 및 디바이스의 저장 공간을 고려하여 가장 적합한 데이터들이 가장 적합한 디바이스에 적재되어 서비스가 제공되도록 다수의 디바이스들에 분산된 데이터를 적절히 보호하고, 다수의 사용자로 콘텐츠가 분배될 때의 네트워크 성능 이슈를 효과적으로 해결하는 서비스 프로토콜과 플랫폼의 개발을 목표로 한다.

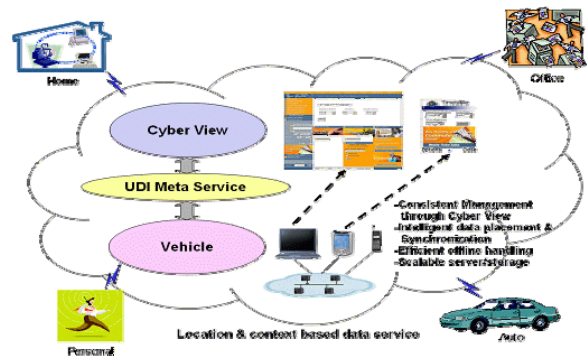


그림 1 UDI (Ubiquitous Data Integration) 서비스 플랫폼

이를 위해 UDI 에서는 사용자의 데이터, 응용, 이들을 수행하는 컴퓨팅 자원의 관리 정책인 논리적 스키마 Cyber View 를 정의하며, 이와 별도로 유비쿼터스 컴퓨팅 환경을 구성하는 물리적인 네트워크, 서버, 휴대 디바이스, 센서, 액츄에이터 등의 물리적 인프라 및 그들의 위치를 Vehicle 로 정의한다. 유비쿼터스 컴퓨팅 환경에서는 응용 및 데이터에 해당하는 Cyber View 가 수행되는 환경인 Vehicle 이 시시각각 변하게 되는데, 이러한 동적 변화에도 불구하고 사용자 및 애플리케이션 개발자들이 항상 일관적이고 효과적인 Cyber View 를 제공받게 하여 손쉬운 애플리케이션 개발 및 관리를 가능하게 한다. 이러한 관리/운용의 편의성을 제공하기 위하여 논리적 스키마(Cyber View)와 물리적인 장치(Vehicle)를 서로 연계하는 UDI 메타 서비스 미들웨어가 개발된다.

III. Cyber View 명세화 도구

Cyber View 는 UDI 의 관리/운용의 편의성을 제공하는 논리적인 스키마로 다음과 같은 명세 방법을 개발한다.

첫째, Event, Location, Context 기반 Behavior 명세 기술개발은 각 Event, Location, Context 별로 필요한 데이터 및 프로그램의 동작을 명세하여 다양한 Vehicle 에서 수행 가능하도록 개발하여 다른 Cyber View entity 와 통신을 위한 인터페이스를 명세하여 협업을 지원하고 기존 애플리케이션 서버 및 모바일 소프트웨어 환경에 구현될 수 있는 연동성을 제공하는 기술이다.

둘째, 데이터 및 관리 정책 명세는 사용자의 의도를 분석하여 필요할 것으로 예측되는 데이터를 디바이스로 pre-fetching 또는 캐싱하기 위한 정책, 데이터 속성, 생명주기, 응용특성, 저장비용을 고려하여 데이터 저장위치를 결정하는 명세이다.

셋째, 데이터 보호 (data protection) 정책은 데이터의 가용성을 높이기 위한 복제 정책 명세 및 복제본 관리 정책 개발, 데이터의 분실과 도난 시 보호를 위해 암호화 지원한다.

넷째, 콘텐츠 분배 정책은 사용자의 콘텐츠에 대한 권한과 분배 설정에 따라 Vehicle 내에 데이터 배분을 지원한다.

다섯째, 디바이스 타입 별로 요구되는 데이터의 속성 명세는 최소 필요 데이터 (응용 동작을 위한 최소

데이터 집합), 선택적인 데이터 (디스플레이, 메모리크기, 네트워크 링크 등)를 Vehicle 의 컴퓨팅 자원에 따라 선택적으로 사용 가능한 데이터 속성을 명세 한다.

이러한 명세 방법은 XML[2] 기반의 스키마로 작성되고, XML 문서 형식으로 표현된다. Cyber View 명세화 도구는 이렇게 잘 정의된 XML 스키마를 읽어 들여 사용자에게 기본적으로 정의된 내용을 보여주고, 쉬운 인터페이스를 통하여 사용자가 쉽게 원하는 작업을 수행할 수 있도록 명세 변경 GUI 를 제공한다. 사용자는 Cyber View 명세화 도구를 통하여 UDI 의 관리/운용의 편의성을 제공 받을 수 있다.

Cyber View 명세화 도구는 Vehicle 의 플랫폼 독립적인 수행환경을 감안하여 JAVA 기반의 응용 애플리케이션[3][4] 형태로 개발되며, 잘 설계된 Cyber View 스키마문서를 기반으로 한 XML 문서를 읽어 들여, 사용자가 원하는 작업이 수행 가능하도록 변경, 추가, 삭제의 과정을 거친 후 최종결과를 저장한다. [그림 1]의 UDI Meta Service 는 정의된 Cyber View 명세를 바탕으로 Vehicle 이 동작하게 하는 Event Handler 의 역할을 수행하여 종합적인 UDI 서비스 플랫폼을 구성하게 된다.

IV. 결론

본 논문은 UDI 서비스 플랫폼 전체 서비스 구조와 이를 위한 명세화 도구 개발에 대하여 언급하였다. 개발된 Cyber View 명세화 도구는 UDI 서비스 플랫폼을 효율적으로 사용할 수 있도록 Event/Location/Context 에 따른 데이터 및 프로그램의 동작을 명세하고, 데이터 배치 및 보호, 콘텐츠 분배 등의 데이터 관리정책, 디바이스에 따라 요구되는 데이터의 속성을 명세하여 다양한 Vehicle 이 명세에 정의된 동작이 수행 가능하게 만드는 데이터와 응용 프로그램의 논리적 스키마를 구성할 수 있는 편의성을 제공한다.

참고문헌

- [1] http://en.wikipedia.org/wiki/Ubiquitous_computing.
- [2] 최범균, 김인희, 김봉수, "자바와 XML", May. 2002
- [3] <http://www.jdom.org/>
- [4] [1] Remote Procedure Calls and Java Remote Method Invocation, IEEE Transactions on Concurrency, Vol.6 Issue 3, pp.5~7, 1998