

유비쿼터스 컴퓨팅 환경에서 PARLAY X를 이용하는 MDA기반의 적응성 있는 문맥인식 서비스

홍 성 준[†]

요 약

본 논문은 유비쿼터스 컴퓨팅 환경에서 PARLAY X기반 서비스 전송 플랫폼상의 MDA(Model Driven Architecture)기반 SCE(Service Creation Environment)를 이용하는 ACS(Adaptive Context-aware Service)에 관하여 서술하였다. 유비쿼터스 컴퓨팅 환경에서는 망 수준에서 문맥인식성과 더불어 적응성이 요구된다. 그러나 기존의 문맥인식성 미들웨어는 적응성에 대한 고려가 부족하다. 그러므로 본 논문의 목적은 유비쿼터스 컴퓨팅 환경에서 문맥인식성과 적응성이 동시에 지원 가능한 PARLAY X기반의 신규 망 서비스 구조 및 PARLAY X API를 개발하고자 하는 것이다. 본 논문에서 제안한 ACS(Adaptive Context-aware Service)는 사용자 주변에 변화하는 문맥 제약조건을 감지하고, 변화하는 문맥 제약 조건에 따라서 적응성 있는 망 서비스를 제공하는 것을 의미하며, ACS를 이용하는 예로 위치와 속도 등의 문맥정보를 인식한 후, 사용자에게 망에서 문맥정보에 따라서 적합한 차별화된 QoS를 지원하는 경우를 보였다. ACS의 구조는 SCE, Adaptive Context Broker, 그리고 PARLAY G/W로 구성되어 있다. SCE는 기존 지능망의 망 서비스와 같은 망 서비스로서 문맥인식성과 적응성을 표현하고 지원하기 위한 망 서비스 개발 환경으로 본 논문에서는 CCL(Context-based Constraint Language)을 이용하였다. Adaptive Context Broker는 SCE의 문맥인식성 및 적응성 표현과 기존 PARLAY G/W사이의 브로커 역할을 한다. PARLAY G/W는 PARLAY X기반의 서비스 전송 플랫폼을 위한 API(Application Programming Interface)를 제공한다.

키워드 : 문맥인식성, PARLAY X, MDA, 유비쿼터스 컴퓨팅 환경

An MDA-Based Adaptive Context-Aware Service Using PARLAY X in Ubiquitous Computing Environments

Hong Sung June[†]

ABSTRACT

This paper describes an Adaptive Context-aware Service (ACS) using Model Driven Architecture (MDA)-based Service Creation Environment (SCE) on PARLAY X based service delivery platform in ubiquitous computing environments. It can be expected that both the context-awareness and adaptation in ubiquitous computing environments will be deployed. But the existing context-aware middleware lacks in considering adaptation. Therefore, the object of this paper is to support the architecture and the Application Programming Interface (API) of the network service for both the context-awareness and adaptation in ubiquitous computing environment. ACS is to provide users with the adaptive network service to the changing context constraints as well as detecting the changing context. For instance, ACS can provide users with QoS in network according to the detected context, after detecting the context such as location and speed. The architecture of ACS is comprised of a Service Creation Environment (SCE), Adaptive Context Broker and PARLAY gateway. SCE is to use Context-based Constraint Language (CCL) for an expression of context-awareness and adaptation. Adaptive Context Broker is to make a role of the broker between SCE and PARLAY G/W. PARLAY G/W is to support API for PARLAY X-based service delivery platform.

Key Words : Context-awareness, PARLAY X, Model Driven Architecture(MDA), Ubiquitous Computing Environment

1. 서 론

최근 유비쿼터스 컴퓨팅 환경에 대한 관심이 증가되고 있

으며, 유비쿼터스 컴퓨팅 환경은 빠르게 변화되는 환경에 적용할 수 있는 유연한(seamless) 응용 프로그램의 요구가 대두되고 있다.

이러한 요구에 부응하여 망 측면에서는 PARLAY[1]라는 개방된 네트워크 서비스 전송 플랫폼 표준이 제정되었다. PARLAY는 1998에 망 오퍼레이터, 망 벤더로 구성되었고 일

※ 본 연구는 2005년도 교내 연구비 지원으로 수행된 과제임
 † 정 회 원 : 여주대학 정보통신과 조교수
 논문접수 : 2005년 2월 7일, 심사완료 : 2005년 3월 29일

반 프로그래머가 네트워크를 사용하거나 부가 가치 서비스를 더 지원할 수 있다는 생각으로 시작되었다. PARLAY는 전화 통신망과 IT기술, 인터넷 그리고 프로그래밍 패러다임을 통합하기 위한 개방된 API(Application Programming Interface)이다. PARLAY 표준화 단체에서는 일반 개발자의 응용 프로그램이 안전하고 실시간으로 통신망 자원을 사용할 수 있도록 해주는 API(Application Programming Interface) 제정 활동을 하고 있다. 특히 PARLAY X는 XML기반의 웹 서비스 기술을 이용하여 네트워크 프로토콜 추상화로 서비스 프로그래밍을 지원한다.

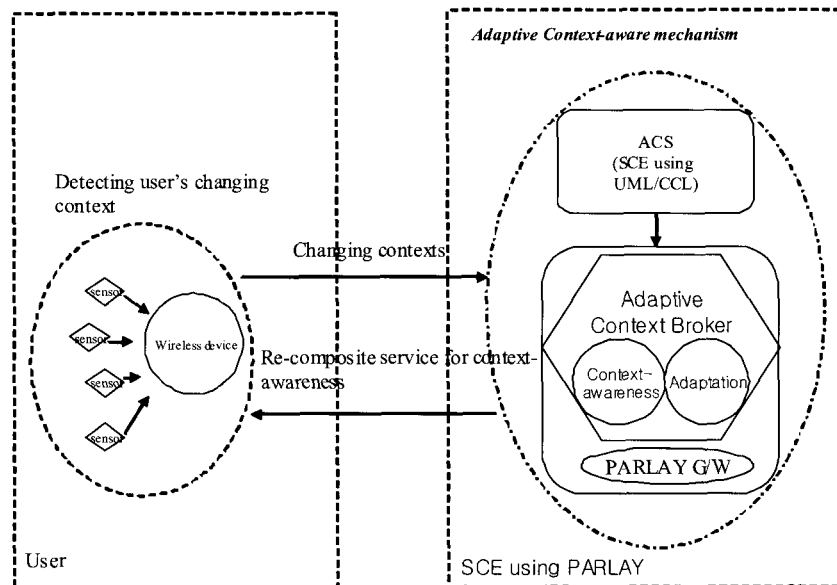
한편, 기술 상황이 지속적으로 변화하기 때문에 특정 기술에 기반된 시스템을 기술 변화에 맞게 통합, 변화, 유지하는 것은 여전한 문제로 남아있다. 이러한 문제점을 해결하기 위해서 OMG(Object Management Group)는 MDA(Model Driven Architecture)[2], [3]라는 개발 패러다임을 제시하였다. MDA는 소프트웨어, 하드웨어, 사람, 비즈니스를 구성하는 기업 환경 시스템의 통합과 상호 운영성에 대한 접근 방법이며, 기업 시스템의 모든 측면을 이해하고 설계하고 동작하도록 하는 프레임 워크를 제공한다. 결과적으로 PARLAY 서비스 전송 플랫폼에서 빠르고 자동화된 네트워크 서비스를 위한 MDA기반의 SCE(Service Creation Environment)[4]를 사용하는 것이 필수적이며, MDA기반의 SCE를 지원하는 도구로는 medini[5] 등이 있다.

유비쿼터스 컴퓨팅 환경에 적합한 네트워크 서비스를 위해서는 문맥인식성과 더불어 적응성 지원이 요구되고 있다. 그러나 기존의 medini와 같은 MDA기반 SCE는 유비쿼터스 환경에서 문맥인식성[6]만 고려하고 적응성[7] 고려는 부족한 실정이다. 문맥인식성을 지원하는 미들웨어로는 RCSM (Re-configurable Context-Sensitive Middleware)[8]이나 WASP (Web Architecture for Service Platform)[9] 등이 있다. 그러나 RCSM은 CORBA기반의 문맥인식 미들웨어로서 단지 미

들웨어 수준에서 문맥인식성 지원에 대해서만 고려하였다. 또한 WASP는 너무 시멘틱 웹 서비스 기술[10], [11]에 의존하여 시멘틱 웹 서비스 기술에 익숙하지 않은 개발자들이 새로운 망 서비스를 개발하는데 많은 어려움이 있다. 기존 문맥인식 미들웨어의 문제점으로 첫째, 유비쿼터스 컴퓨팅 환경에서 문맥인식성과 적응성의 동시 지원에 대한 고려가 부족하다. 둘째, 효율적인 서비스 전송 플랫폼과 문맥인식 언어의 통합에 대한 고려가 부족하다.

그러므로 본 논문의 목적은 문맥인식성과 적응성이 동시에 지원 가능한 PARLAY X기반의 신규 망 서비스 구조 및 PARLAY X API를 개발하고자 하는 것이다. 본 논문의 장점은 첫째, 기존의 PARLAY X에서의 지원 범위를 넘어서는 문맥인식성과 적응성을 표현하고 지원할 수 있다. 둘째, 문맥인식성과 적응성과 같은 지능성을 망에서 제공하므로 제 4세대 무선 통신망이나 BcN(Broadband convergence Network)기반의 차세대 통신 망에서의 기술적인 요구사항을 지원할 수 있다.

본 논문에서 제안한 ACS(Adaptive Context-aware Service)는 사용자에게 변화하는 문맥 제약 조건에 따라서 적응성 있는 망 서비스를 제공하는 것을 의미한다. 한 예로, ACS를 이용한 시나리오에서는 무선 게임을 즐기는 상황에서 위치를 이동하는 경우를 가정하였고, GPS나 센서가 사용자의 위치나 속도 정보를 인식한 후, 그 위치와 속도 정보를 ACS에게 알려주면 ACS는 망 자원 상태를 파악하고 그 지역의 망 자원 상태에 적합한 차별화된 등급의 서비스를 전송하도록 하여 망에서 적합한 QoS (Quality of Service)를 사용자에게 제공한다. ACS의 구조는 SCE와 Adaptive Context Broker, 그리고 PARLAY G/W로 구성되어 있다. SCE는 기존 지능망의 망 서비스와 같은 망 서비스로서 문맥인식성과 적응성을 표현하고 지원하기 위한 망 서비스 개발 환경을 의미한다. 본 SCE는 문맥인식성과 적응성 표현을 위해서 CCL(Context-based Constraint Language)[12]을 이용하였다. CCL은 문맥기반 제



(그림 1) ACS의 개요

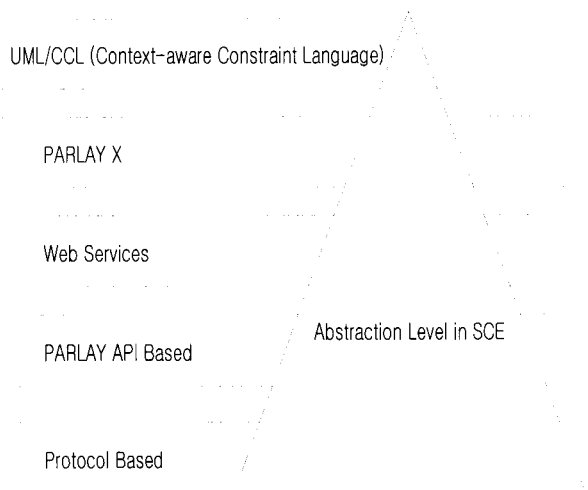
약조건을 표현하기 위한 일종의 UML확장이며, 이 CCL플러그인이 오픈소스 UML 도구에 의해서 지원된다. Adaptive Context Broker는 SCE의 문맥인식성 및 적응성 표현과 기존 PARLAY G/W사이의 브로커 (broker)역할을 한다. PARLAY G/W는 서비스 전송 플랫폼을 위한 API(Application Programming Interface) 제공하는 G/W이고, BcN환경에서 신규 망 서비스 제공을 위한 필수적인 요소 기술로 사용된다.

2. ACS의 설계

2.1 ACS의 개요

(그림 1)은 ACS 의 전체적인 구조를 보이고 있다. ACS의 구조는 UML/CCL를 이용하는 SCE와 Adaptive Context Broker 그리고 PARLAY G/W로 구성되어 있다. Mote[13]라고 불리는 센서나 GPS(Global Positioning System)등으로부터 문맥을 인식한 후 Adaptive Context Broker는 그 문맥을 분석하고 그 문맥에 가장 적합한 서비스를 재구성하여 사용자에게 적응성 있는 서비스를 제공할 수 있다.

(그림 2)는 SCE에서 사용되는 UML/CCL과 PARLAY X의 추상화 수준에 대한 비교를 보이고 있다. CCL은 UML(Unified Modeling Language)의 확장인 OCL(Object Constraint Language)[14]과 유사하며 문맥기반 제약조건을 표현하기 위해 OCL보다 더 추상화된 표현 방법을 사용한다. 여기서 OCL은 UML의 확장이며, UML의 의미(semantics)를 표현하는데 주로 사용된다. PARLAY X는 기존의 PARLAY API보다 더 추상화 표현 방법이며 Web Service기술을 기반으로 하고 있다. PARLAY API Based는 PARLAY X에 비해서 덜 추상화되어 있고, 주로 CORBA, C++ 또는 Java 프로그래밍 언어 등으로 표현되었다. Protocol Based는 거의 추상화가 되지 않은 실제적인 네트워크를 표현한다. PARLAY X는 유비쿼터스 컴퓨팅 환경에서의 문맥인식성에 대한 고려가 부족하다. 그래서 본 논문에서는 문맥기반 제약조건 표현이 가능한 UML/CCL을 기존의 PARLAY X에 적용하였다.



(그림 2) SCE에서의 추상화 수준

2.1.1 ACS의 시나리오

ACS를 사용하는 시나리오는 다음과 같다. 본 시나리오에서는 Mote나 GPS로부터 무선 디바이스에서의 위치나 속도를 문맥으로 가정하였다. 온라인 게임도중에 게임을 계속하면서 사용자는 밖으로 나가기로 결심한다. 비록 무선 디바이스의 주변에 네트워크 자원이 부족하여 스크린상의 화면 등급의 질이 떨어지더라도 서비스의 중단 없이 계속해서 서비스된다. 즉 망 내부적에서 적응성 있게 대역폭 등을 조정하여 서비스가 중단되지 않도록 한다. 그러므로 사용자는 계속해서 무선 디바이스 상에서 게임을 즐길 수 있다.

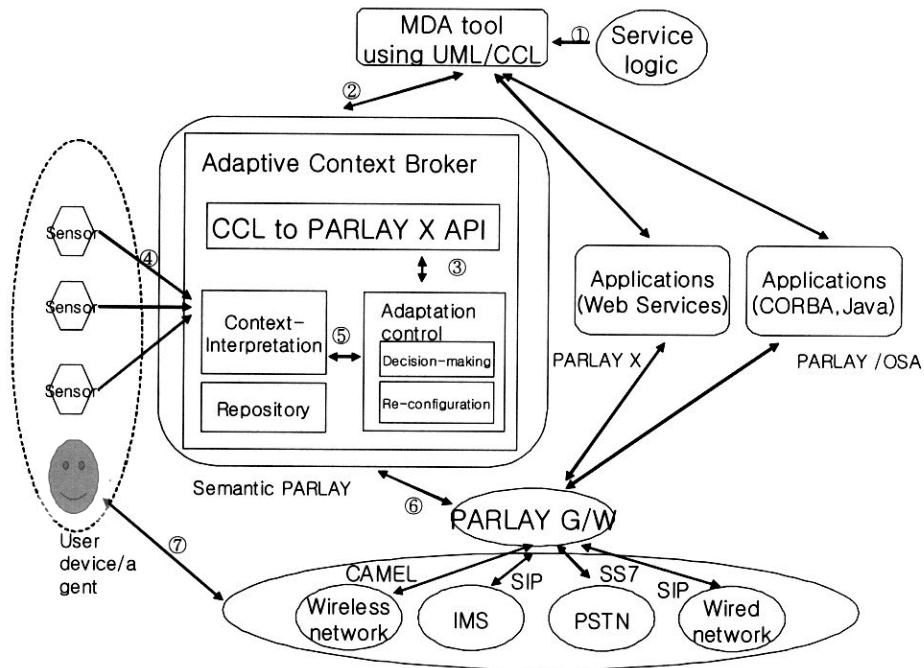
2.1.2 기존 문맥인식 미들웨어와 ACS의 비교

표1은 RCSM와 WASP 그리고 본 논문의 ACS를 비교하였다. RCSM은 CORBA 기반의 미들웨어로서 문맥인식성을 제공한다. WASP는 시멘틱 웹 서비스를 사용하면서 PARLAY G/W와 연동가능 하며, WASP상에 문맥인식성 및 보안을 위한 WSL(WASP Subscription Language)[9]라는 별도 언어가 제안되었다.

정의된 문맥에는 위치(location), 속도(speed), 시간(time), 날씨(weather), 사용자 선호도(User's Preference) 등이 있고, 이러한 문맥은 RCSM, WASP, ACS에서 공통적으로 지원된다. 통신 프로토콜에서 RCSM은 CORBA 기반으로 IIOP(Internet Inter-ORB Protocol)을 지원하고 WASP는 웹 상에서 XML을 기반으로 SOAP(Simple Object Access Protocol)을 사용한다. ACS는 PARLAY를 기반으로 SIP(Session Initiation Protocol), SS7(Signaling System 7), 그리고 MAP(Mobile Application Part)등을 지원한다. 지원 가능한 서비스로 RCSM은 미들웨어상에서 문맥인식 서비스를 지원하고, WASP는 웹 상에서 문맥인식성을 지원한다. 그리고 ACS의 서비스는 PARLAY 기반으로 문맥인식성과 적응성을 지원하는 문맥인식 서비스를 지원한다. 본ACS는 WASP와 유사한 접근방법을 가지고 있지만, 시멘틱 웹 서비스 기술에 독립적이고 통합적인 접근방법인 MDA기반으로 망 서비스를 개발하고 망 내에서 문맥인식성과 적응성을 제공하는 특징을 가진다.

〈표 1〉 기존 문맥인식 미들웨어와 ACS의 비교

	RCSM	WASP	ACS
배경	미들웨어	PARLAY와 시멘틱 웹 서비스를 이용하는 망 플랫폼	PARLAY X상의 MDA기반의 SCE를 이용하는 망 서비스
문맥	위치, 속도, 시간, 날씨, 사용자 선호도(User's preference)	위치, 속도, 시간, 날씨, 사용자 선호도	위치, 속도, 시간, 날씨, 사용자 선호도
통신 프로토콜	IIOP	SOAP	SIP,SS7,MAP
서비스	문맥인식 서비스	문맥인식 서비스	적응성 있는 문맥인식 서비스



(그림 3) ACS의 동작

2.2 ACS의 동작

(그림 3)은 ACS를 위한 동작 구조를 보이고 있다. ACS의 동작은 다음과 같다.

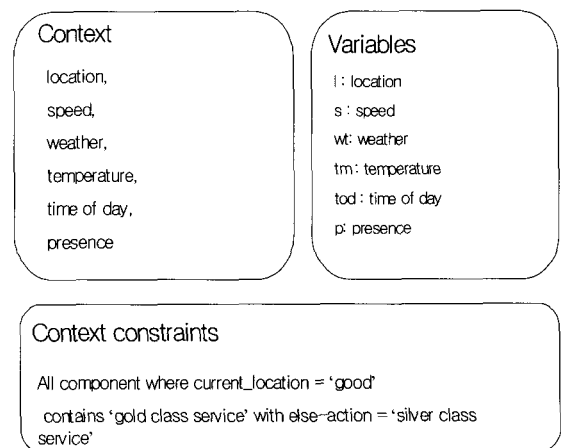
1. MDA기반 SCE에서 서비스 로직 정의와 문맥인식 제약조건을 정의한다.
2. ACS라 명명된 망 서비스 API를 이용한 UML/CCL 정의를 Adaptive Context Broker에게 전달한다.
3. 정의된 UML/CCL은 PARLAY X로 변환된다.
4. Mote라 불리는 많은 센서나 GPS는 Adaptive Context Broker에게 인식된 문맥 정보를 이벤트가 발생하면 전달하는 방식으로 알려준다.
5. Adaptive Context Broker는 인식된 문맥정보를 XML기반의 문맥정보로 변환한 후 문맥에 따라서 적합한 서비스로 재구성한다.
6. Adaptive Context Broker는 XML기반의 문맥정보를 XML기반의 웹 서비스로 변환하고 그것을 PARLAY G/W에 맞게 변환하여 전송한다.
7. PARLAY상의 ACS는 사용자에게 ACS의 결과로 적응성 있는 서비스를 제공한다.

Parlay G/W는 무선망에서는 CAP(CAMEL (Customized Application of Mobile Enhanced Logic) Application Part) 이나 MAP(Mobile Application Part)을 사용하고 IMS(The 3G Multimedia Subsystem)망에서는 SIP(Session Initiation Protocol), PSTN(Public Switched Telephone Network)망에서는 SS7(Signaling System 7)그리고 유선 망에서는 SIP등을 이용한다. 본 논문에서는 Adaptive Context Broker를 이

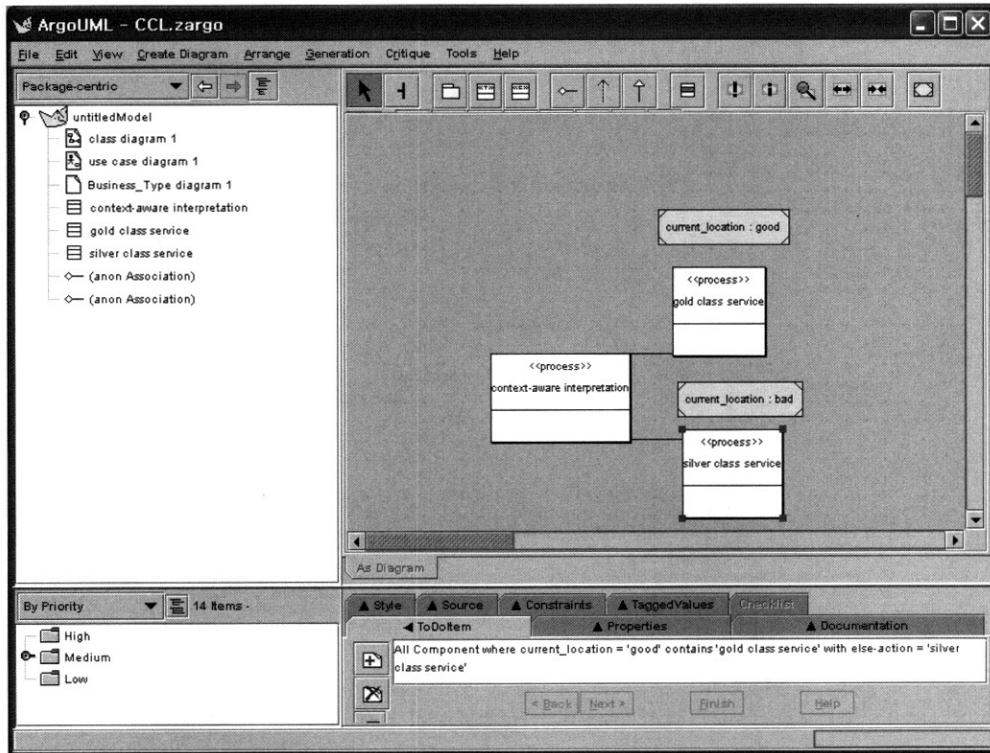
용하는 PARLAY API를 Semantic PARLAY로 명명하였다.

2.3 ACS를 위한 문맥(context) 정의

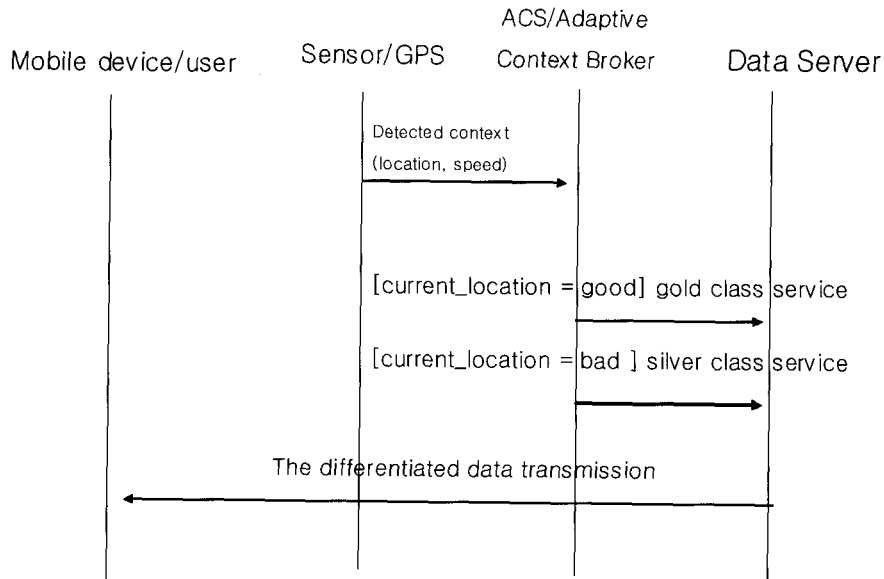
(그림 4)는 ACS를 위한 문맥, 변수, 제약조건을 보이고 있다. 문맥은 위치, 속도, 날씨, 온도 등을 포함한다. 문맥에 대한 (한 변수 *l*은 location, *s*는 speed, *wt*는 weather, *tm*은 time of day, *p*는 presence이다. CCL을 이용한 문맥 제약조건은 다음과 같다. All component where current_location = 'good' constraint 'gold class service' with else-action='silver class service'는 현재 위치가 네트워크 자원이 풍부한 곳이면 gold class service를 제공하고 현재 위치가 네트워크 자원이 부족한 곳이면 silver class service를 제공하라는 의미를 가진다.



(그림 4) 문맥인식을 위한 문 변수, 제약조건



(그림 5) PIM/CCL을 이용하는 서비스 로직과 문맥기반 제약조건의 정의



(그림 6) 시나리오 시퀀스 다이어그램

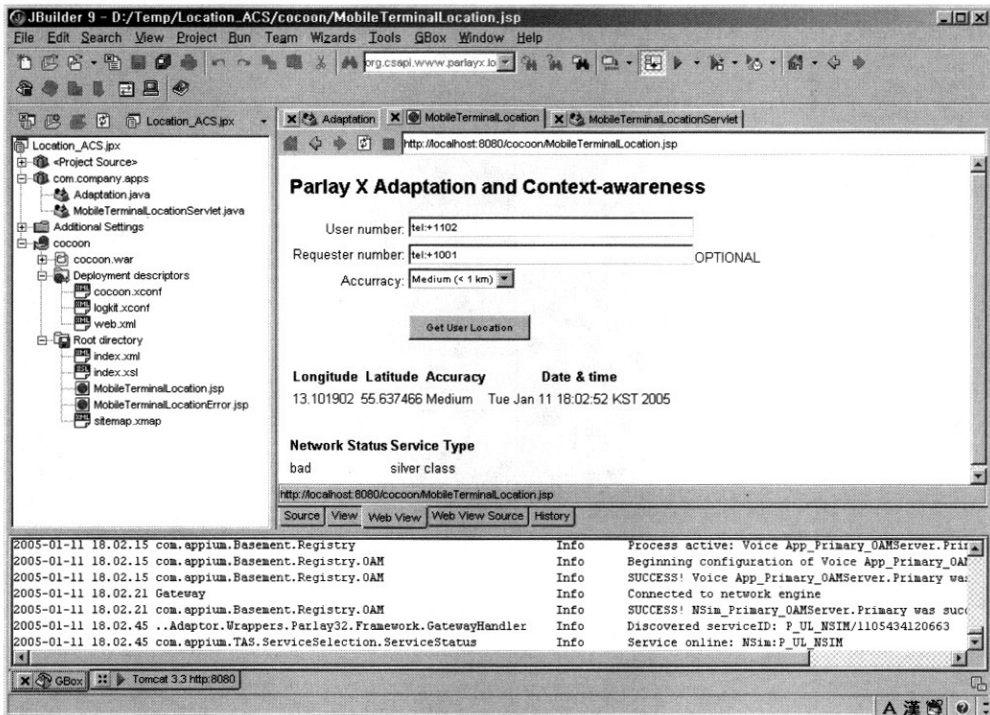
3. ACS의 구현

3.1 PIM (Platform Independent Model)/CCL

(그림 5)는 PIM/CCL을 이용하는 서비스 로직과 문맥기반 제약조건의 정의를 보이고 있다. (그림 5)의 오른쪽 상단에서는 서비스 로직을 그래픽한 표현으로 정의한 것이고 하단은 CCL문법에 맞게 문자적인 표현으로 정의한 것이다. (그림 5)의 서비스 로직의 내용은 (그림 4)에서 언급한 서비스 로직과

동일하다. 서비스 로직을 위한 UML 표현과 더불어 문맥기반 제약조건을 위한 CCL 표현은 ArgoUML[15]도구 및 CCL 플러그인을 이용하였다. CCL은 OCL(Object Constraint Language) [15]보다 추상적 의미 전달이 가능하다.

(그림 6)은 ACS를 위한 시퀀스 다이어그램을 보이고 있다. 센서나 또는 GPS로부터 location과 speed에 대한 문맥을 인식한 후 ACS/Adaptive Context Broker에 알려준다. ACS/ Adaptive Context Broker는 문맥을 분석 하는데 한



(그림 7) PARLAY SDK상의 구현

예로 *current_location* 이 *good* 인지 *current_location* 이 *bad* 인지를 분석한다. 그 후에 Data Server로부터 사용자는 그 문맥에 적합한 서비스를 지원받는다. 한 예로 *current_location*이 *good*이면 *gold class service*를 지원하고 *current_location*이 *bad*이면 *silver class service*를 제공한다.

3.2 PIM/CCL에서 PARLAY PSM(Platform Specific Model)로의 변환

PIM에서 PSM으로 자동 변화 도구는 많이 있지만 PIM/CCL에서 PARLAY PSM으로의 자동 변화 도구가 없어서 본 논문에서는 수동으로 변환하였다.

3.3 PARLAY SDK상의 구현

(그림 7)은 (그림 5)에서 표현된 서비스 로직이 PARLAY X 구현환경에서 동작하는 구현 결과 화면을 보이고 있다. 구현 환경은 Appium사의 GBOX[16]라는 PARLAY X를 지원하는 소프트웨어 개발 도구를 이용하였다. 본 논문에서는 (그림 8)에서 정의된 *networkStatus()*와 *adaptiveServiceType()*이라는 PARLAY X API를 정의하고 그 API를 위한 간단한 응용을 구현하였다. 본 논문에서는 GBOX라고 불리는 PARLAY X지원 시뮬레이터에서 제공되는 지도 상의 x, y 좌표 값을 설정하여 네트워크 지원이 풍부한 지역과 네트워크 자원이 부족한 지역을 구분하여 설정하였다. (그림 7)은 위치 문맥 정보를 얻어서 그 특정 위치의 네트워크 자원이 풍부하면 *gold class service*를 제공하고 그 특정 위치의 네트워크 자원이 부족하면 *silver class service*를 제공하는 것을 보이고 있다.

ACS API

- Context-awareness
 - networkStatus()
- Adaptation
 - adaptiveServiceType()

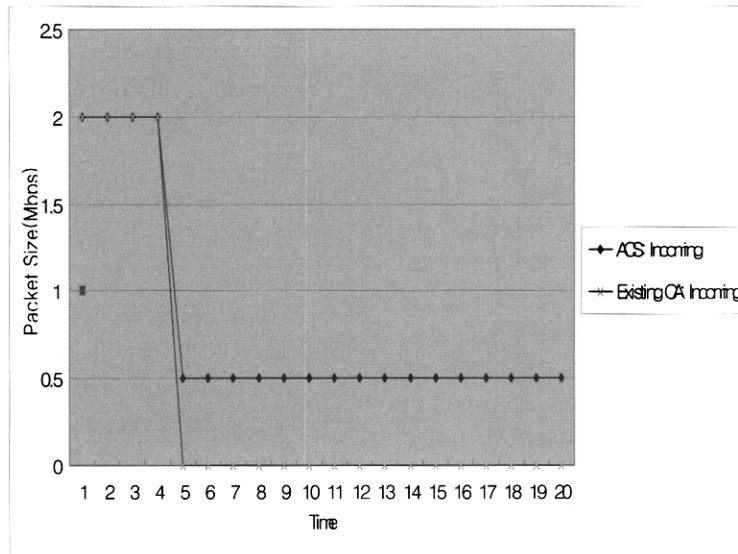
(그림 8) ACS를 위해 정의된 API

(그림 8)은 본 논문에서 ACS를 위해서 정의된 API를 보이고 있다. 문맥인식성을 위한 API로는 *networkStatus()*이고 적응성을 위해서는 *AdaptiveServiceType()*이 있다. *networkStatus()*는 센서나 GPS로부터 문맥정보를 입력받는 기능을 한다. *AdaptiveServiceType()*은 문맥에 대한 정보에 따라서 적응성 있게 서로 다른 서비스를 지원한다.

4. 성능 비교 평가

4.1 특성 비교

<표 2>에서는 기존 문맥인식 미들웨어인 RCSM, WASP, 그리고 ACS를 비교하였다. RCSM은 CORBA를 이용하는 미들웨어로서 주로 문맥인식성을 고려하였다. RCSM은 PARLAY 표준은 지원하지 않고, 적응성에 대한 고려가 부족하고 MDA를 지원하지 않는다. WASP는 시멘틱 웹 서비스 기술을 이용하여 주로 문맥인식성을 고려하였고 일부 적응성 지원이



(그림 9) 성능 비교

가능하다. WASP는 PARLAY표준이 지원되며, MDA를 지원하지 않는다. ACS는 문맥인식성과 적응성을 모두 고려하였다. 그리고 PARLAY 표준을 지원하며UML/CCL을 포함하는 MDA를 지원한다. 또한 ACS에서 Adaptive Context Broker은 SCE에서의 CCL을 이용한 문맥인식 언어와 PARLAY G/W인 서비스 전송 플랫폼 사이에 브로커 역할을 수행함으로써, 결과적으로 문맥인식 언어와 PARLAY G/W를 통합 시키는 역할을 한다.

<표 2> 주된 특성의 비교

	RCSM	WASP	ACS
PARLAY X상의 망 서비스 지원	-	X	X
MDA지원	-	-	X
적응성	-	X	X
문맥인식성	X	X	X
문맥인식언어와 서비스 전송 플랫폼 통합	-	-	X

4.2 성능 비교

RCSM과 ACS가 모두 문맥인식성을 지원하기 때문에 RCSM과 ACS 사이에 적응성에 대한 특징에 대해서 주로 성능을 비교하였고, 성능비교는 ns-2 시뮬레이터를 이용하였다. ns-2에서 성능 실험을 위해 4개의 노드를 설정하였다. 0번 노드는 GPS이고 1번 노드는 데이터 제공 서버이고, 2번 노드는 ACS서비스가 실행되는 서버이고 3번 노드는 사용자 단말기로 설정하였다.

성능 실험을 위해서 0번 노드인 GPS는 사용자의 위치를 감지한 후 2번 노드인 ACS실행 서버에게 알려준 경우, 2번 노드인 ACS 실행 서버는 사용자의 위치 및 그 위치의 네트워크 자원 상태에 따라서 gold class service와 silver class

service등의 서비스 유형을 결정하고 1번 노드인 데이터 제공 서버는 요청되는 서비스 유형에 따라서 데이터를 전송할 때 사용자에게 전송되는 데이터 패킷의 변화량을 측정하는 실험을 하였다. 서비스 유형 변화 지원을 위해서 ns-2를 이용한 Tcl과 C++코드에서 데이터 전송 도중 서비스 유형이 바뀌는 경우를 위한 ServiceTypeChange()메소드를 별도로 포함시켰다. ServiceTypeChange()메소드는 ACS의 성능 실험을 위해서 정의된 메소드이고, 서비스 유형이 데이터 전송 도중에 변경됨에 따라 패킷 크기가 변경되도록 하는 기능을 지원한다.

(그림 9)에서는 성능 비교를 위해서 ACS incoming, Existing CS incoming으로 구성되어 있다. ACS incoming은 ACS를 이용한 경우 3번 노드인 사용자 단말기에서 패킷 크기를 측정된 것이고, Existing CS incoming은 기존 문맥인식성 지원 서비스인 경우 3번 노드인 사용자 단말기에서 패킷 크기를 측정된 것이다. ACS를 의미하는 2번 노드인 ACS 실행 서버가 서비스 유형 변경을 지시하면 1번 노드인 데이터 제공서버는 패킷 크기를 gold class service 인2048에서 silver class service 인 500byte로 조정하면서 서비스를 진행하는 경우, 3번 노드인 사용자 단말기의 패킷 크기 변화를 ACS incoming 그래프에서 보이고 있다. 반면에서 3번 노드인 사용자 단말기의 Existing CS incoming 그래프는 패킷의 크기가 조정되지 않으므로 중간에 서비스가 중단되는 그래프를 보이고 있다. 그러므로RCSM은 적응성에 대한 고려가 부족하기 때문에 망 상황이 변경된 경우 서비스가 중단되었으나 ACS는 변화된 환경에 맞게 서비스가 중단되지 않고 지원되므로 본 ACS가 적응성을 지원한다는 점을 보인다.

5. 결론 및 향후 연구

본 논문은 망에서 문맥인식성과 적응성을 지원하기 위한 PARLAY X기반의 망 서비스 구조와 PARLAY X API를 제

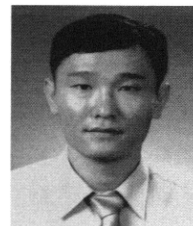
안하였고 위치와 속도라는 문맥정보를 인식하여 망에서 그 문맥정보에 따라서 망에서 QoS를 사용자에게 제공하는 경우를 보였다. 본 PARLAY X상의 ACS는 기존 문맥 인식 지원 미들웨어 보다 망 상에서의 적응성과 같은 지능적인 서비스를 지원할 수 있는 장점이 있다. ACS는 향후 보다 다양한 문맥정보에 따라서 망에서 더욱 지능적인 서비스를 제공할 수 있을 것이며, 이러한 망에 지능성을 제공하려는 시도는 제4세대 무선망이나 차세대 네트워크에 필수적인 기술적인 요소가 될 것으로 보인다. 또한 본 논문의 망 서비스를 위한 구조와 신규 PARLAY API는 PARLAY X표준에 반영될 것으로 기대된다. 향후 연구로는 유비쿼터스 환경에서 시멘틱 웹 서비스와 PARLAY를 통합하여 망에서의 상황인식성[17], [18] 지원에 관한 연구가 필요하다. 더불어 시멘틱 웹 서비스 기술과 PARLAY의 효율적인 통합을 위해서 MDA를 이용한 통합 방안에 관한 연구가 필요하다.

참 고 문 헌

[1] PARLAY home page, available at www.parlay.org.
 [2] O. Kath, T. Magedanz, R. Wechselberger: "MDA-based Service Creation for OSA/Parlay within 3G beyond Environments," First European Workshop on Model Driven Architecture with Emphasis on Industrial Application, March 17-18, 2004. University of Twente, Enschede, The Netherlands.
 [3] Aniruddha Gokhale et al. "CosMIC: An MDA Generative Tool for Distributed Real-time and Embedded Component Middleware and Applications," Proceedings of the OOPSLA 2002 Workshop on Generative Techniques in the Content of Model Driven Architecture, Seattle, WA, November, 2002.
 [4] Bernhard Stenffen, Tiziana Margaria, Andreas Claben, Volker Bruan, "A Constraint-Oriented Service Creation Environment," 2nd International Conference on Practical Application of Constraint Technology, London (UK), April, 1996.
 [5] IKV++ Technologies AG: The medini tool chain, available at <http://www.ikv.de/pdf/mediniWhitePaper.pdf>.
 [6] C. Efstathiou, K. Cheverst, N. Davies and A. Friday, "An architecture for the effective support of adaptive context-aware applications," **Proceedings of 2nd International Conference in Mobile Data Management (MDM'01)**, Hong Kong, Springer, Vol. Lecture Notes in Computer Science Volume 1987, pp.15-26, January, 2001.
 [7] S. Pokraev et. al., "Context-aware Service," Technical Report WASP/D2.3, Telematica Instituut, Ericsson, November, 2003, available at <https://doc.telin.nl/dscgi/ds.py/Get/File-27859/>.

[8] S. S. Yau, F. Karim, Y. Wang, B. Wang, and S. Gupta, "Reconfigurable Context-Sensitive Middleware for Pervasive Computing," *IEEE Pervasive Computing*, joint special issue with *IEEE Personal Communications*, 1(3), pp.33-40, 2002.
 [9] Pareica Dockhorn Costa et. al. "Toward a Service Platform for Mobile Context-Aware Applications," *First International Workshop on Ubiquitous Computing (IWUC 2004)*, Portugal, April, 2004.
 [10] Harry Chen, Tim Finin, Anupam Joshi, "Semantic Web in the Context Broker Architecture," In *Proceedings of PerCom 2004*, Orlando FL., March, 2004.
 [11] Norman M. Sadeh, "A Semantic Web Service Environment for Context-Aware Mobile Services," *Wireless World Research Forum Conference*, Stockholm, September, 2001.
 [12] Felix Bubl, "The context-based constraint language CCL for component," Technical report, Technical University Berlen, available at www.CoCons.org, 2002.
 [13] L. Girod, J. Elson, A. Cerpa, T. Stathopoulos, N. Ramnathan, D. Estrin, "EmStar: a Software Environment for Developing and Deploying Wireless Sensor Networks," in the *Proceedings of USENIX General Track 2004*.
 [14] J. Wing, and Kleppe, A., "OCL: The Constraint Language of the UML," *JOOP*, May, 1999.
 [15] ArgoUML home page, available at <http://argouml.tigris.org>.
 [16] Appium Homepage, SDK supporting PARLAY X, available at <http://www.appium.com/>.
 [17] Rudiger Gartmann, Agenes Voisard, Norbert Weibenberg, "Situation-Aware Service Supply," 2002 Workshop on Generative Techniques in the Content of Model Driven Architecture, Seattle, WA, November 2002.
 [18] Hung Q. Ngo, Sungyoung Lee, "Middleware for context-aware ubiquitous computing," *정보처리학회지*, 제11권 제6호, pp.56-75, 2004.

홍 성 준



e-mail : sjhong@mail.yeojoo.ac.kr
 1991년 경원대학교 전자계산학과(공학사)
 1993년 건국대학교 일반대학원 전자계산학과(공학석사)
 1998년 건국대학교 일반 대학원 컴퓨터공학과(공학박사)

1993년~1999년 한국통신 연구소 전임연구원
 1999년~현재 여주대학 정보통신과 조교수
 관심분야: 유비쿼터스 컴퓨팅, 통신 미들웨어, PARLAY