

모바일을 위한 UML 2.0 기반의 아키텍처 모델링 언어 프레임워크

박용우⁰ 김현성 전태웅
고려대학교 전산학과
{ywpark⁰, saron1, jeon}@selab.korea.ac.kr

UML 2.0 based ADL Framework for Mobile Application

Yongwoo Park⁰ Hyunsung Kim Taewoong Jeon
Dept. Computer Science, Korea University

요 약

Acme와 같은 ADL(Architecture Description Language) 들은 컴포넌트 기반 시스템의 아키텍처를 정형적으로 명세할 수 있는 장점을 가지고 있다. 하지만 날로 다각화되는 개발 과정 상의 이해관계들을 다 포용하지 못하고, 아키텍처에 특화된 별도의 표기 형식을 익혀야 하는 부담이 있어 아키텍처를 명세하는 언어로서 정착되지 못하고 있다. 반면 UML(Unified Modeling Language)은 범용 모델링 언어이며 소프트웨어 개발의 전 과정에 일관된 표기형식과 폭넓은 지원도구들을 제공하고 있어 소프트웨어 개발을 위한 사실상의 표준 언어로 자리잡고 있다. 이에 따라 지금까지 UML을 이용하여 아키텍처를 모델링하기 위한 연구들이 많이 진행되어 왔다. 특히 UML에서 표현수단이 미흡한 아키텍처의 핵심 개념들을 명시적으로 표현할 수 있도록 UML의 확장 메커니즘을 사용하여 UML을 특화하는 연구 결과들이 많이 소개되고 있으나 특화된 영역의 아키텍처를 기술하기에는 부족하다. 본 논문에서는 최근에 활발히 개발되고 있는 모바일 애플리케이션의 특징중의 하나인 device의 제약사항을 QoS로 정의하고 UML 확장메커니즘을 이용하여 모바일을 위한 아키텍처 모델링 언어 프레임워크를 제시하고 있다.

1. 서론

1.1. 소프트웨어 아키텍처 모델

소프트웨어 아키텍처는 상위 추상화 수준에서 시스템을 구성하는 연산 또는 데이터 저장 요소(컴포넌트)들과 연결요소(커넥터), 그리고 이들의 합성에 적용된 합성 규칙과 제약조건을 나타내는 합성 패턴을 포함한다[1]. 이러한 아키텍처를 잘 기술한 모델은 시스템의 전체적인 구조를 가시화하여 시스템에 대한 이해를 돕고, 요구사항과 구현을 연결하는 역할을 할 수 있으며, 컴포넌트와 이들이 통합될 수 있는 프레임워크에 대한 재사용성을 지원한다[2]

1.2. 아키텍처 기술 언어 (Architecture Description Language)

ADL은 아키텍처 수준에서 개념적인 프레임워크를 제공하고 해당 소프트웨어를 특징화하기 위한 표기법으로 볼 수 있다. 대부분의 ADL들은 소프트웨어 아키텍처의 서로 다른 측면에 초점을 맞추어 개발되었다.

Adage, Aesop, Darwin, Rapide, SADL, Wright 등이 그러한 대표적인 언어들이다. 이와 같은 ADL들에 대한 연구는 주로 소프트웨어 아키텍처를 연구하는 학계에서 진행되고 있다. ADL은 틀셋과 함께 제공되는 경우가 많아 그래피컬한 환경에서 설계가 가능하고, 코드 생성, 실행 시 모니터링, 이상 검출, 성능 분석 등이 가능하다. 이러한 많은 ADL들의 근저에 깔려 있는 기본적인 핵심 개념들을 추려내고 이들을 통합하고자 하는 목적으로 만들어진 차세대 ADL로서 ACME 가 있다. ACME에서는 많은

ADL들이 공통적으로 가지고 있는, 아키텍처 기술에 있어 기본이 되는 개념들을 컴포넌트, 커넥터, 시스템 구성, 속성, 제약조건, 스타일(Styles)로 정의 하였다.

1.3 UML

ACME는 아키텍처의 서로 다른 측면을 다루는 ADL 들을 통합할 수 있는 프레임워크를 제공한다. 하지만 ACME는 이에 통합할 각각의 ADL이 지원하는 아키텍처에 특화된 별도의 표기형식을 익혀야 하는 부담이 있어 아키텍처를 표현하는 언어로서 널리 자리잡지 못하고 있다. 이에 따라 ACME가 지원하는 개념들을 UML로 표현하는 방법들이 연구되고 있다. 이러한 접근 방식은 UML과 같은 표준 언어를 이용해서 아키텍처가 기술되기 때문에 아키텍처에 대한 이해가 용이하고 일관성 있게 이루어지며, 기존에 사용되고 있는 툴의 지원을 받을 수 있다. 이러한 장점들로 인해 UML 표기를 이용해 아키텍처를 표현하는 연구들이 진행되고 있다.

UML은 범용 모델링 언어로서 아키텍처 기술에 완전히 특화된 개념을 제공하지는 않는다. UML 2.0에서는 이전 버전에서 미흡했던 아키텍처 모델링에 유용한 개념들이 많이 추가되었지만 여전히 UML로 명시적인 표현이 어려운 아키텍처의 핵심 개념들이 존재한다. 그러므로 UML을 기반으로 아키텍처를 정확하게 모델링하기 위해서는 UML을 이에 맞게 확장할 필요가 있다.

1.4 본 논문의 구성 내용

본 논문에서는 모바일 애플리케이션의 특징을 QoS로 정의하고 UML 확장메커니즘을 이용하여 모바일 상거래 아

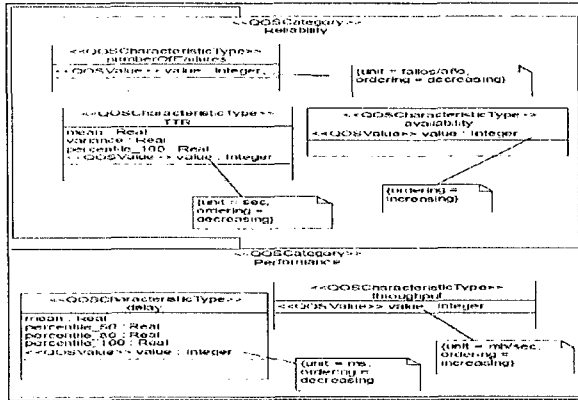


그림1 QoS Category 패키지

키텍처 모델링 언어를 제시하고 있다. 우선 UML을 통한 모델링 언어에 관한 연구를 영역에 독립적인 요소와 영역에 고유한 핵심 요소를 각각 Generic ADL과 DS-ADL(Domain-Specific ADL)로 정의하고, 서비스의 품질을 보장받기 위한 기술 및 개념인 QoS (Quality of Service)에 대해 이야기 한다. 그리고 모바일 상거래 영역을 주 도메인으로 잡고 모바일 애플리케이션의 특징을 QoS로 정의하고 DS-ADL에 QoS를 가미한 MC-ADL을 정의한다. 마지막으로 MC-ADL을 통한 식당예약시스템의 예제를 보여준다.

2. 관련 연구

UML을 통한 모델링 언어에 관한 연구는 영역에 독립적인 요소와 영역에 고유한 핵심 요소를 각각 Generic ADL과 DS-ADL(Domain-Specific ADL)로 정의하고, 정의된 DS-ADL을 사용하여 해당 영역에서 재사용 가능한 요소들을 프레임워크 아키텍처 모델로 구축하는 2가지가 있었다.

2.1. Generic-ADL

영역에 독립적인, 모든 아키텍처 모델이 갖는 공통적 개념들의 어휘를 제공하는 Generic ADL을 UML의 확장 메커니즘을 이용하여 UML 2.0 프로파일로 정의한다. Generic-ADL은 그 자체만으로 아키텍처 모델링이 가능한 수준을 유지하도록 일반적인 아키텍처 모델링을 위한 핵심 개념들을 모두 포괄하도록 정의한다.

2.2. DS-ADL

위에서 이야기한 Generic-ADL을 확장하여 영역에 종속적인 아키텍처 상의 개념을 표현하는 DS-ADL(Domain-Specific ADL)을 정의한다. DS-ADL은 영역에 종속적인 아키텍처 설계 요소들의 어휘 사전을 제공한다. 이를 적용하여 Framework Architecture를 정의할 수 있다. 이것은 특정 영역에 공통적으로 나타날 수 있는 설계 요소간 제약조건과 시스템 구성(Configuration)을 표현한 것으로 문제 영역의 아키텍처 설계 지식을 표현한 것이라 할 수 있다. 해당 영역에 속한 특정 목표 시스템의 아키텍처는 DS-ADL을 사용하여 설계된다. 해당 영역을 M-Commerce로 잡고 정의하였으나 모바일 영역에 대해 미흡한 점이 아직 많이 있다.

2.3 QOS (Quality of Service)

QOS (Quality of Service)는 서비스의 품질을 보장받기

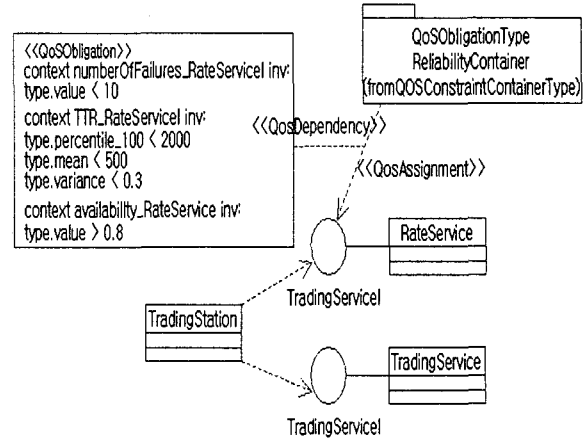


그림2 QOS의 모델 수준에서 적용한 예

위한 기술 및 개념이다. 처리능력, 전송지연, 오류빈도, 연결의 실패 확률, 정확성 및 신뢰성 등의 매개변수를 사용하여 데이터 전송의 특성을 기술한다. 최근에는 시스템의 기능적인 요소보다 비 기능적인 요소의 영향을 중요하게 다루고 있다. QoS는 시스템이 제공하는 기능적인 측면에서 사용자의 만족하는 수준을 설명하는 비 기능적 요소의 집합이며 시스템이 놓여진 환경에 잘 적응되기 위한 주요한 측면이다[3].

2.3.1 QOS(Quality of Service) 메타 클래스

UML 확장 메커니즘을 이용하여 QOS 핵심 개념들을 UML 메타모델로 확장하여 QOSCore Class Diagram을 정의 하였다[3].

2.3.2 QOS(Quality of Service)의 모델 수준에서 표현

QOS(Quality of Service)의 모델 수준에서 적용하는 형태에 대해서도 예제를 통하여 보여준다 [3].

그림 1은 QOSCharacteristic을 나타내는 기본단위라고 할 수 있는 QOSCharacteristicType의 그룹으로 정의된 모델링 요소들의 인스턴스 Package인 QOSCategory 이다. Reliability와 Performance에 대한 QOSCharacteristicType의 인스턴스를 정의하고 있다. 그림 2는 QOSCategory에서 구성된 QOSCharacteristicType으로 인터페이스의 조건이 OCL로 표현된 것을 컴포넌트의 인터페이스에 연결함으로써 인터페이스에 다른 커넥터나 컴포넌트가 연결되기 위한 조건을 표기하여 사용자 모델 수준에서 적용하는 부분이다.

3. 모바일 상거래 영역을 위한 UML2.0에 기반한 아키텍처 모델링 언어의 정의

3.1 MC-ADL의 메타모델 아키텍처

모바일 상거래 영역의 비 기능적 제약 사항을 아키텍처 모델링에 표현하기 위해서 QOS를 사용하여 제약사항을 모델링에 표현한다. 그림 3은 영역에 핵심 요소를 정의된 DS-ADL에서 모바일 상거래 영역의 특징들을 QoS를 가미해서 MC-ADL(M-Commerce ADL)로 정의한다.

QOSCore는 UML메타모델에서 스테레오타입으로 확장했다. QOSCore를 기반으로 모델 수준에서 사용될 QOSCharacteristicType을 정의하고 이것들의 모음을 사전(Library)화하여 QOSLibrary를 구성한다.

프레임워크 아키텍처는 모바일 상거래 영역에서 공통적으로 나타날 수 있는 설계 요소간 제약 조건과 시스템 구성을 표현한 것으로 MC-ADL을 기반으로 하는 목표 시스템 아키텍처 설계의 기본 설정을 표현하고 있다. 모바일 상거래 영역의 목표 시스템 아키텍처는 MC-ADL을 사용하고 프레임워크 아키텍처 모델과 QOSLibrary 모델을 참조 또는 확장하여 설계한다.

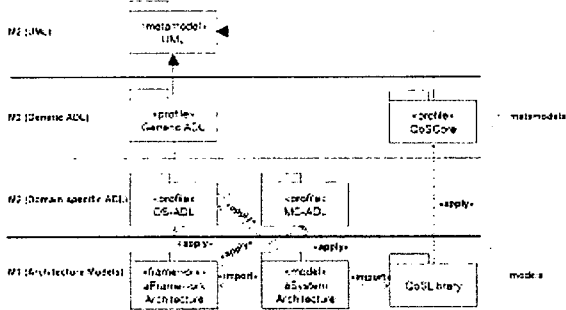


그림3 MC-ADL 메타모델 아키텍처
QOS에 적용되는 모바일 영역의 특징 중의 하나인 해당 Component에 설치될 Device의 정보크기/필수정보여부/크기변경에 따른 개인정보 저장위치 판단 3가지만 가지고 이야기를 하겠다. 그림 4는 이 모바일 상거래 영역의 특징을 기반으로 세 개의 CharacteristicType인 Size, Personality, Importance를 정의하여 <<QOSCategory>> Capability 패키지를 구성하였다.

<<QOSCategory>> Capability 패키지는 QOSCharacteristicType의 인스턴스 모임으로 모바일 상거래 영역의 목표 시스템 아키텍처 모델링을 위한 QOSLibrary이다.
<<CharacteristicType>> Size는 CharacteristicType의 인스턴스로 Attribute Value의 타입으로 <<QOSValue>>를 갖는다. Attribute Value의 값은 Integer이다. Attribute isVariable은 변수의 여부를 판단하는 기준으로 Boolean 값을 타입으로 갖는다.

<<CharacteristicType>> Personality와 <<CharacteristicType>> Importance도 마찬가지로 CharacteristicType의 인스턴스로 Attribute Value의 타입으로 <<QOSValue>>를 갖고 Attribute Value의 값은 Integer 타입이다

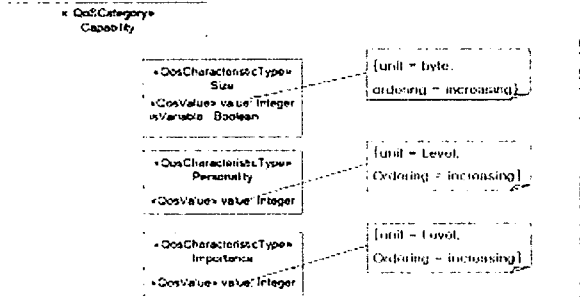


그림4 QOS CharacteristicType 인스턴스의 Capability 패키지
3.2 MC-ADL를 사용한 아키텍처 모델링의 예
그림 5는 ReservationAgent 컴포넌트의 포트와 포트에

연결된 인터페이스의 구조를 정의한 예를 보여준다. 인터페이스에 QOS를 정의하여 인터페이스에 연결되는 커넥터와 다른 컴포넌트의 접속 조건을 설정하였다. 인터페이스는 Provided 인터페이스와 Required 인터페이스로 두 가지 형태로 포트를 통해서 표현된다.

<<QOSObligation>> 정의된 QOSCharacteristic의 인스턴스는 (Size, Personality, Importance)_ISystemAccess이다. Size, Personality, Importance를 각각의 타입으로 갖는다. Size_ISystemAccess는 ISystemAccess 인터페이스에 허용된 정보의 크기를 설정하고 있다. 그리고 Personality_ISystemAccess는 개인화 정도를 표현하고 Importance_ISystemAccess는 인터페이스에 허용된 정보의 중요한 정도를 표현하고 있다. 내용은 예제의 표현을 위하여 임의로 type.value < 1000 등과 같이 값을 갖는 형태로 설정하였다.

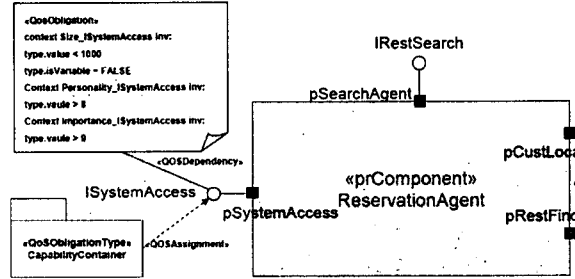


그림5 컴포넌트의 포트와 인터페이스에 QOS 적용

4. 결론 및 향후 연구

본 논문에서는 최근에 활발히 개발되고 있는 모바일 애플리케이션의 특징 중의 하나인 device의 제약사항을 QoS로 정의하고 UML 확장메커니즘을 이용하여 모바일을 위한 아키텍처 모델링 언어 프레임워크를 제시하였다. 향후에는 모바일의 특징 중 하나인 context-awareness로 MC-ADL을 보강하겠다.

참고문헌

- [1] Mary Shaw, David Garlan, Software architecture: perspectives on an emerging discipline, Prentice-Hall, Inc., Upper Saddle River, NJ, 1996
- [2] David Garlan: Software Architecture: a Roadmap, ICSE 2000, 22nd International Conference on Software Engineering, Future of Software Engineering Track, June 4-11, 2000, Limerick Ireland, ACM, 2000.
- [3] J. J. Asensio, "A UML Profile for QOS Management Information Specification in Distributed Object based Applications," presented at HP Open View University Association 7th Plenary Workshop (HPOVUA 2000), Santorini, Greece.
- [4] UML 2.0 Superstructure, 3rd Revision, OMG document ad/03-04-01, Object Management Group, 2003, www.omg.org/cgi-bin/doc?ad/03-04-01.
- [5] 김경래, 노성환, 전태웅, "UML2.0에 기반한 아키텍처 모델링 언어", Joint Workshop on Software Engineering Technology (KSEJW-2004), August 26-27, 2004, pp. 125-134