모바일 응용 S/W를 위한 ADL 정의+

김희열, 곽재경, 전태웅
고려대학교 전산학과
e-mail:{hy210,jk_kwak,jeon}@korea.ac.kr

Defining an ADL for Mobile Application Software

Hee-Yul Kim, Jae-Kyung Kwak, Taewoong Jeon
Dept of Computer Science, Korea University

요 약

모바일 단말기에 탑재되는 적응형 모바일 응용 SW의 아키텍처 모델링을 지원하는 ADL을 UML 프로파일로 제시하였다. 제시된 모바일 ADL은 정적 아키텍처의 표현 요소를 기본으로 제공하고 그 위에 동적 아키텍처의 표현 요소들이 추가된 구조로 정의되었다.

1. 서론

최근의 S/W 개발 방식은, 여러 응용 영역들에 걸쳐서 또는 특정 영역에서 공통적인 부분들을 추상화, 계층화, 모듈화하여 핵심 S/W 자산(core S/W assets)으로 구축하고 이를 조직적으로 재사용함으로써 특정 응용과 플랫폼에 맞는 S/W 제품들을 효율적으로 생산하는 방향으로 발전하고 있다. 상위 추상화 수준에서의 S/W의 구조와 행위가 잘 정의된 아키텍처의 중요성이 더욱 커지고 있다. 이에 따라 소프트웨어의 아키텍처를 명확하게 설계하고 이를 중심으로 소프트웨어를 효과적으로 분석, 구현, 유지관리하는 아키텍처 중심의 개발 방법들이 많이 연구되어 왔다.

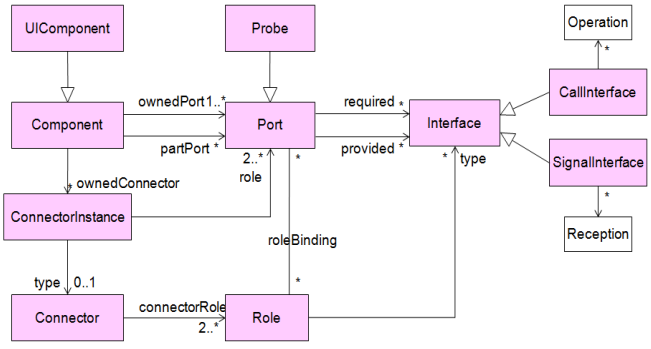
본 논문은 모바일 단말기에 탑재되어 실행되는 모바일 응용 SW의 정적 아키텍처와 동적 아키텍처를 기술하는데 필요한 표현 수단을 제공하는 UML 기반의 아키텍처 기술언어를 제안한다. 정적 아키텍처는 시스템에서 실행 시 구조가 변하지 않는 부분의 아키텍처이다. 동적 아키텍처는 실행 환경의 변화에 적응하여 동적으로 재구성되는 아키텍처이다[1,2].

현재 모델링 언어 표준으로 자리잡은 UML이 소프트웨어의 분석, 설계에 많이 사용되고 있다. 아키텍처 수준의 설계에서도 UML의 사용이 보편화되고 있다. UML은 아키텍처 수준에서 핵심적인 개념들을 모델 요소들에 직접 반영하는 방향으로 진화하여, 현재 버전인 UML 2.1[3]에서는 아키텍처의 핵심 개념들이 많이 추가, 보완되었다. 하지만 UML 2.1도 여전히 아키텍처의 주요 측면들을 모두 명시적으로 표현하는데 충분하지 않다[4,5]. 더욱이 UML을 그대로 사용하여 모바일-아키텍처를 표현하는 것은 도메인 종속적인 아키텍처의 핵심 개념들과 UML의 범용 모델링 개념들 사이의 차이로 인한 어려움이 따른다. 이에 본 연구에서는 모바일 ADL을 UML을 모바일 응용 영역에 맞게 특화한 UML profile로 정의하였다.

먼저 UML로부터 특정 응용 영역에 독립적인 아키텍처 요소들을 추려내고(generic ADL), 이를 모바일 응용 영역에 적합한 핵심 요소들로 특화하여 모바일 ADL을 정의한다. Generic ADL은 선행 연구에서 정의하였다[6]. 본 논문에서 정의된 모바일 ADL은 정적 아키텍처의 표현 요소를 기본으로 제공하고 그 위에 동적 아키텍처의 표현 요소들이 추가된 구조이다. 정의된 모바일 ADL은 특정 모바일 응용 SW의 아키텍처를 직접 모델링하거나 여러 모바일 응용 SW의 개발에 재사용 가능한 아키텍처 요소들을 프레임워크 아키텍처 모델로 구축하는데 사용한다.

2. 제안된 모바일 ADL의 메타모델

본 논문이 제안하는 모바일 ADL의 핵심 모델 요소들과 이들 사이의 관계를 보여주는 메타모델은 (그림 1)과 같다.



(그림 1) 모바일 ADL의 메타모델

+ 본 연구는 정보통신부 및 정보통신연구진흥원의 IT 신성장동력핵심기술개발사업의 일환으로 수행하였음. [2007-S032-01, 다중 플랫폼 지원 모바일 응용 SW 개발환경 기술 개발]

Component - 독립된 실행 요소이면서 합성의 단위를 나타내는 Classifier이다. 시스템에 조립 부품으로 독립적으로 사용될 수 있도록 외부와의 상호작용에 필요한 인터페이스들은 포트를 통해 정의되어 있고 그 밖의 구성물들은 내부에 은폐되어 있다. UML 2의 컴포넌트에는 없는 실행 시 런타임 플랫폼에 의해 동적 재구성이 가능한 개념이 추가되었다.

Port - Component의 property로서, 컴포넌트와 외부환경 사이 또는 컴포넌트와 내부 구성 요소들 사이의 연결점을 나타낸다. Probe는 동적 재구성이 가능한 컴포넌트들의 연결 관계를 재설정하기 위해 런타임 플랫폼이 컴포넌트의 상태를 제어하고 감시하는데 사용되며 이에 필요한 인터페이스가 미리 정의된 포트이다.

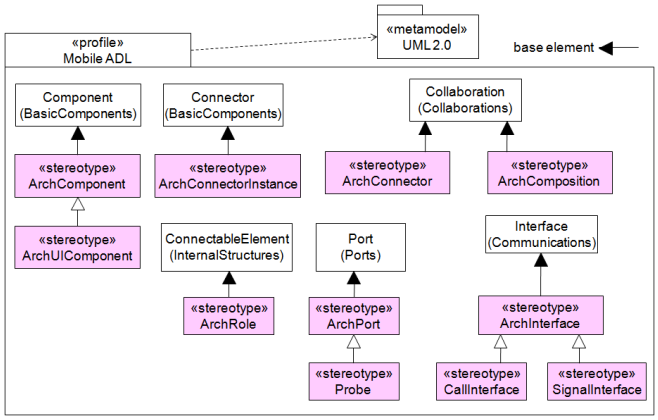
Interface - 컴포넌트들 사이의 상호작용에 사용되는 Operation 또는 Signal들의 집합으로 정의되는 인터페이스이다.

Connector - 컴포넌트들 사이의 연결 관계 유형을 나타내는 Classifier로서, 연결되는 포트들의 역할과 타입을 정의한다. UML 2의 Connector는 인스턴스 수준의 개념이지만, 본 모바일 ADL의 Connector는 커넥터 인스턴스들의 타입을 정의하는 개념으로 지위가 격상되었다. 또한, 시스템 실행 시 런타임 플랫폼에 의해 동적으로 재설정되는 연결 관계 유형을 나타내는 동적 커넥터의 개념도 표현이 가능하도록 의미가 확장되었다.

ConnectorInstance - 컴포넌트들의 상호작용을 위한 포트 사이의 연결을 나타내는 인스턴스 수준의 커넥터로서 Connector를 타입으로 가질 수 있다. ConnectorInstance가 타입을 갖는 경우, 타입으로 정의된 Connector에 설정된 역할(connector role)에 따라 상호작용하는 포트들의 연결 관계를 나타낸다.

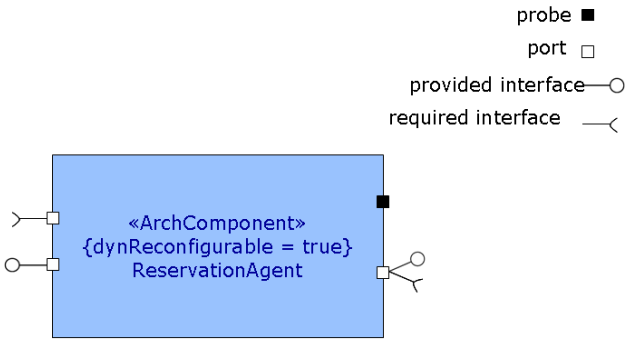
3. 모바일 ADL의 UML 2 프로파일

2장에서 설명한 모바일 ADL 메타모델의 핵심 모델 요소들을 UML 메타클래스들의 스테레오타입으로 정의한 UML 프로파일은 (그림 2)와 같다.



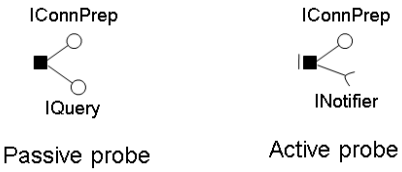
(그림 2) 모바일 ADL의 UML 2 profile

ArchComponent는 한 개 이상의 ArchPort를 가지며 ArchPort는 Provided interface와 Required interface를 가질 수 있다. 이러한 인터페이스는 미리 정의되어야 한다. 동적 재구성 가능 여부를 나타내기 위해 dynReconfigurable property를 가지며, tag로 표현이 가능하다(그림 3). dynReconfigurable ArchComponent는 Probe가 있어야 한다.



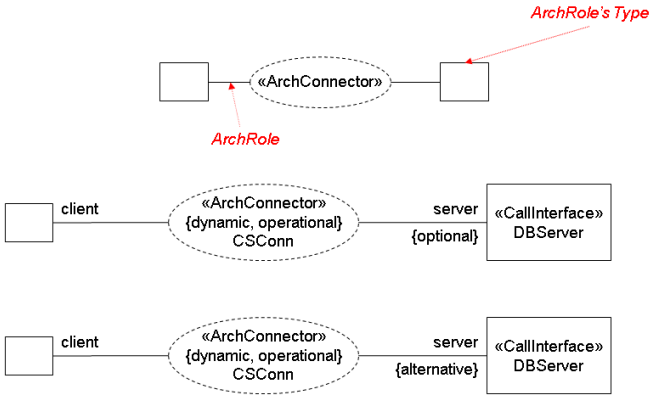
(그림 3) ArchComponent의 표기

Probe는 동적 재구성을 위해 동적 런타임 플랫폼과 통신을 하는 포트로서 ArchPort로부터 특화된다. Probe는 미리 정의된 두개의 인터페이스가 있다. 하나는 런타임 플랫폼과 통신하여 연결을 준비(connection preparation)하기 위한 인터페이스이고, 다른 하나는 런타임 플랫폼이 동적 재구성되는 ArchComponent의 상태 질의(state query) 혹은 ArchComponent가 런타임 플랫폼에게 상태 변화 알림(state change notification)을 위한 인터페이스이다. 연결 준비를 위한 인터페이스는 provided CallInterface로 정의된다. Passive probe는 상태 질의를 위한 provided CallInterface를 갖는다. Active probe는 상태 변화 알림을 위한 required SignalInterface를 갖는다.



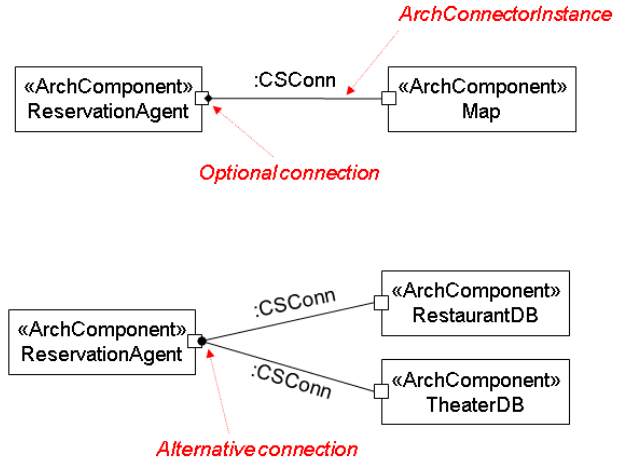
(그림 4) probe의 인터페이스

ArchConnector는 UML 2의 Collaboration 메타클래스의 스테레오타입으로 정의된 커넥터 요소를 나타낸다. ArchConnector에서 합성 패턴에 참여하는 role들의 의미를 제약하여 ArchRole로서 정의한다. ArchRole은 role binding에 의해 ArchPort에만 대응되도록 제약한다. 그리고 동적 재구성 개념을 지원하기 위해 connector role에 동적으로 연결될 포트의 선택 방식(항상 연결, 조건적 연결, 여러 개 중 택일 연결)과 동적 연결이 허용되는 시점(로드시점, 인스턴스 생성 시점, 시작 시점, 실행 중)을 지정할 수 있다. (그림 5)는 이러한 ArchConnector의 표기 예를 보여 준다.



(그림 5) ArchConnector의 표기

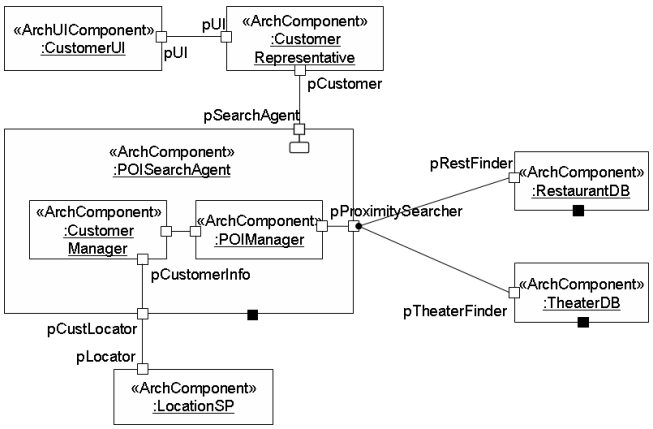
ArchConnectorInstance는 UML 2 Connector의 스테레오타입이다. 이렇게 정의된 ArchConnectorInstance는 자신에 연결되는 ConnectableElement의 유형이 포트로 한정되며 ArchConnector를 타입으로 가질 수 있다. (그림 6)은 (그림 5)의 CSConn 커넥터를 타입으로 갖는 ConnectorInstance의 예를 보여 준다.



(그림 6) ArchConnectorInstance

4. 모바일 ADL로 기술된 아키텍처 모델 예

(그림 3)은 본 연구의 모바일 ADL로 표현한 POI(Point of Interest) Search 응용의 샘플 아키텍처 모델을 보여준다. CustomerUI는 고객의 인터페이스를 나타내는 컴포넌트로 고객이 주변의 관심 장소들의 정보를 얻기 위하여 모바일 기기를 이용하여 접속하는 인터페이스를 나타낸다. CustomerRepresentative는 고객이 원하는 정보를 고객에게 전달하는 기능을 갖는 컴포넌트로 POISearchAgent에게 고객이 원하는 서비스(POI Search)를 전달한다. 고객과 POI 정보를 관리하는 POISearchAgent 컴포넌트가 고객의 위치 정보를 LocationSP 컴포넌트에 요청한다. RestaurantDB와 TheaterDB 컴포넌트는 각각 극장과 식당의 정보를 가지고 있는 컴포넌트이다. 이 두개의 DB 컴포넌트는 고객이 원하는 POI의 유형에 따라 동적으로 택일 재구성된다.



(그림 7) 모바일 ADL 모델 예 : POI Search

5. 결론

본 연구는 [7]에 명세된 다중 플랫폼 지원 모바일 응용 SW개발환경 요구사항을 만족하기 위하여 수행되었다. 이를 위해 본 논문에서는 실행 환경에 맞게 동적으로 재구성되는 적응형 모바일 응용 SW의 아키텍처 모델링을 지원하기 위한 아키텍처 기술 언어를 제안하였다. 제안된 ADL은 아직 실제 적응형 모바일 응용 SW의 개발에 사용되지 아니한 초기 연구 상태이다. 효용성 검증을 위해 본 모바일 ADL을 사용한 아키텍처 모델 생성기와 아키텍처 모델 검증기를 지원 도구로 개발할 예정이다.

참고문헌

[1] V. Grassi, R. Mirandola, and A. Sabetta, "A UML Profile to Model Mobile Systems", UML 2004, LNCS 3273, Springer-Verlag, 2004, pp. 128-142

[2] H. Gomaa and M. Hussein, "Software Reconfiguration Patterns for Dynamic Evolution of Software Architectures", Proceedings of the 4th. Working IEEE/IFIP Conference on Software Architecture (WICSA'04), June 12-15, 2004, pp. 79-88

[3] UML 2.1.1 Superstructure Specification, formal/07-02-05, Object Management Group, 2007

[4] M. Bjorkander and C. Kobryn, "Architecting Systems with UML 2.0", IEEE Software, Vol. 20, No. 4, July-August 2003, pp. 57-61

[5] J. Ivers, P. Clements, D. Garlan, et al., "Documenting Component and Connector Views with UML 2.0", Technical Report, CMU/SEI-2004-TR-008, April 2004

[6] 노성환, 김경래, 전태웅, 윤석진, "UML 2.0 기반의 Generic ADL 정의", 정보과학회 논문지: 소프트웨어 및 응용, 제33권 제2호, 2006년 2월, pp. 167-185

[7] MOPAD 개발팀, 다중 플랫폼 지원 모바일 응용 SW 개발환경(MOPAD) 요구사항 명세서, 버전 1.0, 2007. 8