

# 이종 콘텐츠 소프트웨어의 동적 조합을 위한 안드로이드 플랫폼 기반 프레임워크

한예슬\*, 장태관\*, 이병수\*

\*인천대학교 정보통신대학원 컴퓨터공학과

\*인천대학교 대학원 컴퓨터공학과

\*인천대학교 컴퓨터공학과

e-mail : pusikes@incheon.ac.kr, hinet007@dreamwiz.com

## A Framework based on Android Platform for Dynamic Composition of Different Contents Software

Yeseul Han\*, Tae-Kwan Jang\*, Byungsoo Lee\*

\*Dept. of Computer Science & Engineering, University of Incheon

### 요 약

소프트웨어에 대한 사용자 요구가 점차 다양화하고 복잡화함에 따라 단일 시스템을 단일 어플리케이션 형태로 구축하기보다 여러 개의 단위 기능의 조합을 통해 시스템을 구축할 필요가 높아지고 있다. 이를 위해 여러 모델이 제안되어 왔으나, 구현에 있어 대부분 웹 서비스 기술을 사용하고 있어, 분산 환경이 아닌 모바일 플랫폼에서의 로컬 실행 환경에서는 적합하지 않았다. 또한 개별 서비스를 독립적으로 사용하기 힘들거나 내용상의 의존성이 생겨 특정 서비스에 의존하게 되는 문제 등의 한계가 있었다.

본 연구에서는 시스템의 구성 콘텐츠를 각기 독립된 소프트웨어로 개발하여 표준 데이터 공개를 통해 이들을 동적으로 조합하여 시스템을 구축하는 안드로이드 플랫폼 기반 프레임워크를 설계하고 구현하였다. 본 프레임워크는 웹 서비스를 이용하여 서비스 프로세스를 합성하는 대신 일종의 데이터 공유를 통해 시스템 내 구성요소들을 결합시킴으로써, 컴포넌트로서의 소프트웨어 합성 방법에 다양성을 부여할 것으로 기대된다.

### 1. 서론

SOA 서비스는 시스템 내의 각 소프트웨어 기능을 분해하여 연동할 수 있어 다양한 벤더에서 개발한 서비스 기능 조합을 가능하게 하고, SaaS 는 네트워크를 통한 렌탈 서비스로서의 소프트웨어뿐만 아니라 소프트웨어의 기능 중 사용자가 필요한 기능들만을 선택하여 사용할 수 있게 하는 개념을 포함하는 추세에 있다. 그리고 CBD 는 개발 단계에서의 컴포넌트 사용뿐만 아니라 최종 사용자 개발까지 포함하는 더 넓은 의미로 확대되어 오고 있다. 이는 소프트웨어 수요의 증가와 사용자 요구의 다양화와 함께, 개발 비용 감소와 기간 단축뿐만 아니라 사용자가 직접 소프트웨어들을 동적으로 조합하여 시스템을 구축할 필요성 또한 증가하고 있음에 기인한다[1][2].

이러한 모델 아키텍처에 따라 조합형 시스템을 구현하는 데 있어 의존성 문제가 대두된다. 의존성 문제는 컴포넌트 혹은 서비스의 조합을 통한 개발에서 뿐만 아니라 객체지향 소프트웨어 개발 전반의 중요한 고려 사항으로, 클래스 모듈 간의 높은 의존성은 재사용성, 테스트 용이성 등의 소프트웨어 품질 요소에 악영향을 주어, 형상 관리를 어렵게 만들고 유지

보수 비용을 증가시킨다[1][3].

본 연구에서는 시스템을 구성하는 기능 요소들을 시스템 내에서의 의미에 따라 콘텐츠별로 분리하여, 각 콘텐츠를 그 자체로 완전한 하나의 소프트웨어로서 각각 독립적으로 개발한 후, 해당 콘텐츠에 대해 정의된 표준 공개 데이터를 중앙의 중개자에게 제공하여, 사용자에게 따라 각 소프트웨어를 그대로 사용하거나 혹은 최종 사용자가 직접 이종 콘텐츠 소프트웨어들을 조합하여 의미상으로 결합함으로써 새로운 시스템을 구축할 수 있도록 하는 소프트웨어 모델을 제안한다. 그리고 제안한 모델을 안드로이드 플랫폼 기반의 프레임워크로 설계 및 구현하고, 프레임워크를 적용하여 소프트웨어를 개발하고 평가한다.

실험 결과, 소프트웨어들을 개별적으로 사용하거나 혹은 다양한 조합의 연동을 통하여 동적으로 시스템을 구축하여 사용하는 데 있어서, 그들 사이에 형식적인 의존성뿐만 아니라 의미상의 의존성 역시 존재하지 않는 것으로 나타났고, 각 소프트웨어 프로그램 소스 코드 내에서의 패키지간 의존성 또한 같은 시스템을 종래의 통합적 개발 방식에 따라 개발하였을 때에 비교하여 완화된 결과를 도출하였다.

## 2. 관련연구

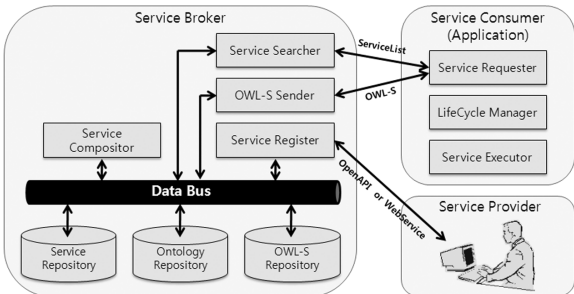
### 2.1. SOA

서비스는 어플리케이션이나 다른 서비스가 외부에서 그 서비스를 찾을 수 있고, 공개된 인터페이스를 통해 접근하며, 주로 메시지 기반의 비동기 커뮤니케이션 방식으로 사용되는 로직의 집합으로서의 프로세스이다. 서비스는 기본적으로 서비스가 요청하고 반환하는 데이터로 설정할 수 있다. SOA(Service-Oriented Architecture)는 이러한 서비스를 기반으로 어플리케이션이나 서비스를 구축하도록 하는 설계 방법 모델 혹은 아키텍처이다. SOA의 구성 요소들은 서로 느슨하게 연결되어 있고 독립적이다[4][5].

그러나, XML과 WSDL 등으로 대표되는 기술적 표준 인터페이스의 정립에도 불구하고, 비즈니스 레벨 표준의 미정립과 난립 등의 문제로 인하여, 서비스 지향 아키텍처의 기본 정신과는 달리 많은 경우 벤더들이 협력 개발에 동의한 이후에 협의된 표준에 맞추어 서비스를 제작하게 되어, 이러한 경우 상대 패키지에 대한 상호 의존성이 없는 서비스간에도 실질적으로 내용상의 높은 의존성이 생기게 된다는 문제가 있다.

### 2.2. SOA 기반 프레임워크

김진한 외 1인이 제시한 프레임워크[6]는 웹 서비스와 OpenAPI를 결합하여 서비스의 동적 합성을 위한 방법을 제공한다. 그림 1은 [6]에서 제시하는 SOA 기반 서비스 합성 프레임워크를 보인다.



(그림 1) SOA 기반 서비스 합성 프레임워크[6]

서비스 합성에는 매쉬업이 사용되는데, 특정한 기능을 수행하는 일련의 프로세스의 조합의 수행을 각 서비스에 위임하여 결과를 얻는 방식으로 합성이 이루어지게 된다. 표 1은 프레임워크의 특성을 간략하게 보여 준다.

<표 1> 비교 프레임워크의 특성

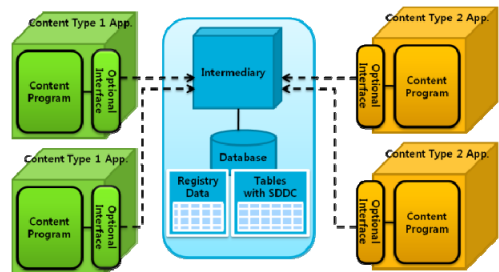
비교 영역	특성
동작 환경	네트워크를 통하여 통신하는 분산 환경
플랫폼 기술	웹 서비스와 OpenAPI
합성 방식	서비스 프로세스 합성
서비스 발견성	서비스 명세, 온톨로지, OWL-S을 통한 서비스 검색에 의함

## 3. 제안 모델과 프레임워크 설계 및 구현

### 3.1. 표준 데이터 공개를 통한 이중 콘텐츠 소프트웨어의 동적 조합 모델

어플리케이션이 중앙의 중개자에 자신의 정보를 등록하고 중개자는 이 정보를 참조하려는 다른 어플리케이션들에게 정보를 제공하여, 콘텐츠 그룹에 대해 정의된 데이터 표준을 통해 어플리케이션간 연동이 이루어지게 하는, 최종 사용자가 직접 어플리케이션들을 선택하고 조합하여 전체 시스템을 구축할 수 있는 모델을 제안한다. 이 모델에서 각 어플리케이션은 독립된 하나의 콘텐츠를 제공하며, 그 자체로 완전한 시스템이면서 또한 더 큰 시스템의 일부이다.

이 모델에서 SDDC(Standard Data Definition for specific Content domain; 특정 콘텐츠 도메인에 대한 표준 데이터 정의)를 통해, 동적으로 조합된 시스템의 구성 어플리케이션이 동종 콘텐츠 유형에 속하는 다른 어플리케이션으로 교체되어도 아무런 의미상의 의존 문제가 발생하지 않도록 할 수 있다. 이 때 SDDC는 콘텐츠 종류에 따라 정의된, 연동 시스템을 위한 비즈니스 영역의 표준 공개 데이터를 의미하며, 서로 다른 벤더가 개발한 어플리케이션이라도 자체 표준에 더하여 이 공통 표준을 추가적으로 따르도록 한다. 각 콘텐츠 그룹에 대한 SDDC를 준수하는 데이터를 통해 어플리케이션들이 조합되게 함으로써 느슨한 결합을 통한 의존성 제어뿐만 아니라 의미상의 의존성 또한 제어되도록 할 수 있다. SDDC의 예로서, 교육용 기능성 게임 시스템의 학습 콘텐츠의 경우 학습에 따른 점수, 학습 시간, 성적, 스케일 등이 포함될 수 있다. 그리고 서로 다른 콘텐츠 타입의 어플리케이션들을 연계하는 역할을 중앙의 중개자가 담당한다. 이 중개자는 저장 영역에 각 어플리케이션의 SDDC 데이터뿐만 아니라 어플리케이션의 등록 데이터를 관리함으로써, 한 어플리케이션이 다른 어플리케이션을 직접적으로 참조하지 않도록 하여 어플리케이션 사이의 의존성을 제어한다. 여기서 등록 데이터는 어플리케이션의 식별자와 콘텐츠 도메인 그룹 정보 등을 포함한다. 그림 2는 SDDC와 어플리케이션 등록을 통한 이중 콘텐츠 소프트웨어의 동적 조합 모델을 나타내고 있다.



(그림 2) SDDC와 어플리케이션 등록을 통한 이중 콘텐츠 소프트웨어의 동적 조합 모델

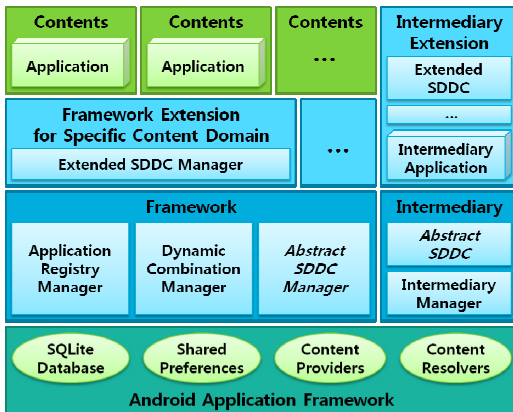
### 3.2. 프레임워크 설계 및 구현

#### 3.1 절의 모델을 바탕으로, 안드로이드 플랫폼을 기

반으로 프레임워크를 설계하고 구현하였다.

프레임워크의 구조는 그림 3 과 같이 기본 프레임워크를 바탕으로 특정 콘텐츠 도메인의 SDDC 가 필요할 때 프레임워크를 확장하여 이를 기반으로 어플리케이션을 개발하도록 되어 있다.

프레임워크는 크게 어플리케이션 개발을 위한 기본 프레임워크 패키지와 중개자 어플리케이션을 위한 패키지 구성되어 있다. 기본 프레임워크에서 어플리케이션이 공통적으로 사용하는 등록 기능과 동적 조합 기능을 어플리케이션 등록 매니저와 동적 조합 매니저 패키지가 각각 담당한다. 이들은 어플리케이션과 중개자의 상호 작용을 위하여 어플리케이션에 선택적으로 추가되는 인터페이스이다. 그리고 SDDC 와 관련한 처리를 위한 추상 SDDC 매니저 패키지가 있다. 추상 SDDC 매니저 패키지를 확장하여 확장된 프레임워크를 정의할 수 있는데, 이 프레임워크를 기반으로 어플리케이션이 개발된다. 한 편, 중개자 어플리케이션을 위한 프레임워크에는 중개자 매니저와 추상 SDDC 정보 패키지가 있다. 이 또한 마찬가지로 특정 SDDC 에 대하여 확장할 수 있고, 동적 조합 시스템을 위해서 한 개 이상의 확장된 SDDC 정보 모듈과 연계된 중개자 어플리케이션을 구현하여야 한다.

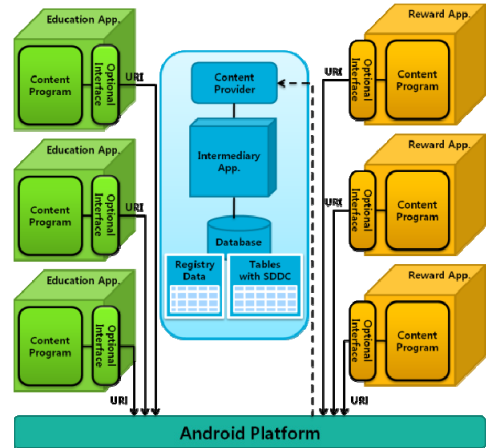


(그림 3) 제안 모델의 안드로이드 프레임워크 구조

안드로이드 플랫폼은 보안상의 이유로 타 어플리케이션에서 생성하고 관리하는 데이터베이스에 접근하는 것을 허용하지 않고, 공용 데이터 저장 영역 또한 존재하지 않으므로, 동적 조합 시스템을 위한 어플리케이션의 등록 데이터 정보와 SDDC 를 준수하는 데이터 원본은 중개자 어플리케이션에 의해 관리되고 중개자의 Content Provider 에 의해 REST 추상화를 거쳐 캡슐화되어 서비스되도록 설계되었다.

또한 프레임워크를 적용한 안드로이드 어플리케이션이 중개자의 Content Provider 와 통신할 때, 특정 중개자 어플리케이션 패키지에 의존하지 않고 URI 를 사용하여 중개자에 접근한다. 이는 어플리케이션과 중개자 사이의 의존성을 완화시키며, 또한 SOA 프레임워크의 데이터 프레임워크가 수행하는 추상화된 형태의 통합 데이터 스키마 제공 기능과 유사하게, 데이터 통합 및 DB 구성 방식에 유연성을 제공한다. 그림 4 는 교육용 기능성 게임 시스템이라는 특정 컨

츠 도메인에 대한 제안 프레임워크 적용시의 이중 콘텐츠 어플리케이션의 동적 연동 모형을 나타낸다.



(그림 4) 특정 콘텐츠 도메인에서의 동적 연동 모형

표 2 는 2.2 절의 프레임워크와 비교한 제안 프레임워크의 특성을 나타낸다. 분산 환경에서 네트워크를 통해 여러 웹 서비스를 식별하는 환경이 아닌, 단일 안드로이드 기기 내에 설치되고 등록된 어플리케이션들을 식별하여 Content Provider 를 통하여 연동하는 동작 환경을 갖는다. 합성 방식은 SDDC 규격의 데이터 공개와 참조를 통하여 간접적으로 소프트웨어간의 의미상의 연동이 이루어지는 방식을 취한다. 연동 소프트웨어 발견성은 서비스 중개자에 등록된 어플리케이션의 등록 정보의 콘텐츠 도메인 그룹 정보를 통해 충족된다.

<표 2> 제안 프레임워크의 특성

비교 영역	특성
동작 환경	네트워크 통신을 고려하지 않는 로컬 환경
플랫폼 기술	안드로이드 플랫폼
합성 방식	표준 데이터 공개를 통한 간접적인 소프트웨어 합성
연동 소프트웨어 발견성	서비스 중개자에 등록된 어플리케이션 등록 정보의 콘텐츠 도메인 그룹 정보에 의함

#### 4. 평가

##### 4.1. 프레임워크를 통한 어플리케이션 개발

프레임워크를 적용하여 안드로이드 어플리케이션을 개발하였다. 교육용 기능성 게임 시스템의 구현을 위하여 시스템들을 분석한 결과, 공통적으로 학습을 위한 콘텐츠와 교육학적인 보상을 위한 콘텐츠로 구성됨을 확인하였다. 이에 따라 학습 콘텐츠 도메인에 대한 SDDC 를 정의하고, 프레임워크를 확장하였다. 이 확장된 프레임워크를 기반으로 학습 콘텐츠 어플리케이션인 토익 RC Part5 학습(Edu1), 세계사 학습(Edu2), 초등 산수 학습(Edu3) 어플리케이션을 구현하고, 또 기존 시스템 내에서 보상 콘텐츠에 해

당하는 슬롯 머신(Rew1), 펜타고 보드 게임(Rew2), 스토쿠 게임(Rew3) 어플리케이션을 구현하였다.

그림 5 는 상기 6 개 어플리케이션들 중 일부의 화면이다. 토익 학습 어플리케이션인 Edu1 에서 학습하여 얻은 포인트를 구현된 동적 조합 매니저 패키지의 UI 를 통해 슬롯 머신 어플리케이션인 Rew1 에서 연동하여 코인으로 이용할 수 있다.



(그림 5) 프레임워크를 적용한 어플리케이션 화면

4.2. 의존성 완화를 통한 시스템 디자인 개선

최종 사용자가 이들을 런타임 시에 조합하여 시스템을 구축하는 실험을 수행하였다. 표 3 은 이 실험을 통해 나타난 시스템을 구성하는 어플리케이션 간의 연동 결과이다. 이를 통해 조합 시스템 내의 어플리케이션간 의존성을 평가할 수 있다.

<표 3> 동적 조합 시스템 구성시 어플리케이션간의 연동 실험 결과

(0: 정상 작동)

	Rew1	Rew2	Rew3	미연동
Edu1	0	0	0	0
Edu2	0	0	0	0
Edu3	0	0	0	0
미연동	0	0	0	-

표 3 의 결과에서 나타난 것처럼, 프레임워크를 적용한 어플리케이션은 외부 패키지와 연동할 때 특정 어플리케이션 패키지에 의존하지 않고, 프레임워크를 적용한 어플리케이션이라면 어디에나 연동할 수 있는 것이 확인되었다. 또한, 패키지 인터페이스상의 의존성이 완화되었을 뿐만 아니라 의미상의 의존성 또한 완화되어, 런타임 시 문제 없이 동작하고 연동 어플리케이션의 교체에 대해 강인성을 갖는 것으로 나타났다.

4.3. 프로그램 코드 품질 향상

표 4 는 유사한 기능을 수행하는 1 개 시스템을 구축하는 데 있어서, 기존의 통합적 개발 방식을 통해 구현된 프로그램인 Edu1\*Rew1 과 프레임워크를 적용하여 분리 개발한 프로그램인 Edu1, Rew1 의 품질을 코드 품질 분석 도구 ‘SourceMonitor’와 ‘JDepend’를 사용하여 분석하고 비교한 것이다.

클래스의 수와 패키지의 수로 대표되는 프로그램 규모 면에서는 각각 평균 53.4%, 55.0%가 감소하였다. B.Boehm 의 COCOMO 모델에서는 소프트웨어 개발 비용이 소프트웨어 규모에 비선형적으로 비례한다고 하였으므로, 어플리케이션 규모의 경량화는 개발 비

용을 규모의 감소폭 이상으로 절감시켜 줄 것으로 예상된다[3]. 그리고 패키지의 평균 의존성은 평균 0.4 만큼 감소하여 23.5%의 감소폭을 보였다.

<표 4> 시스템 구성 콘텐츠별 분리 개발에 따른 코드 특성 변화

(A.Ce: 패키지의 평균 의존성)

코드 품질 분석 도구	SourceMonitor		JDepend
	Classes	Packages	A.Ce
Edu1*Rew1 (구 시스템)	29	10	1.7
Edu1	15	4	1.0
Rew1	12	5	1.6
Average (Edu1,Rew1)	13.5	4.5	1.3
평균 감소량	15.5	5.5	0.4
평균 감소비	53.4%	55.0%	23.5%

5. 결론

본 논문에서는 시스템의 구성 콘텐츠들을 각기 독립된 어플리케이션으로 개발하고 이들을 사용자가 직접 동적으로 조합하여 시스템을 구축할 수 있는 모델과 안드로이드 플랫폼 기반 프레임워크를 설계하고 구현하였다. 이러한 조합형 시스템은 단순히 개별 서비어나 어플리케이션의 개발과 버전 관리의 부담을 감소시킨다는 이점 이외에도, 콘텐츠 유형별로 전문화된 개발 벤더의 개발이 가능하고 다양한 비즈니스 모델의 적용이 가능하다는 점에서 이점이 있다. 이러한 모델은 상대적으로 중소규모 벤더 비율이 높은 모바일 개발 업계에서 활용성이 높을 것으로 기대된다.

참고문헌

- [1] 임윤선, 김명, 정승남, 정안모. “컴포넌트 재사용을 지원하는 컴포넌트 모델 및 프레임워크”, 정보과학회논문지: 소프트웨어 및 응용 제 34 권 제 12 호, 한국정보과학회, 2007.
- [2] 송영재, 김귀정, 변정우, 서영중, 최한용, 한정수. “소프트웨어공학”, 이한출판사, 2004.
- [3] 최은만. “소프트웨어 공학”, 정익사, 2007.
- [4] 신혜원, 윤수진. “SOA 와 CBD 를 이용한 웹서비스 시스템 설계 및 구현”, 한국정보과학회 가을 학술 논문발표집 Vol.31, No.2, 2004.
- [5] Thomas Erl. “Service-oriented Architecture”, Prentice Hall, 2005.
- [6] 김진한, 이병정. “웹 서비스와 OpenAPI 를 이용한 SOA 기반 동적 서비스 합성 프레임워크”, 정보과학회논문지: 소프트웨어 및 응용 제 36 권 제 3 호, 한국정보과학회, 2009.
- [7] Nicolai M. Josuttis. “SOA in Practice”, O’Reilly, 2007.
- [8] Michael J. Carey. “Declarative Data Services: This Is Your Data on SOA”, IEEE International Conference on Service-Oriented Computing and Applications, 2007.