

# REST 프로토콜 기반의 API 선별 기법 및 Open API 자동 합성 방안

김 상 일\*, 김 화 성<sup>o</sup>

## API Selection and Automatic Open API Composition Method Based on REST Protocol

Sang-il Kim\*, Hwa-sung Kim<sup>o</sup>

### 요 약

웹 2.0의 발전과 스마트 디바이스의 보급은 제한적인 정보와 평면적인 서비스에 맞춤형된 기존의 웹서비스를 탈피하여 사용자가 원하는 다차원적인 정보와 다양한 서비스가 융합된 형태의 매쉬업 서비스를 탄생시키는 요인이 되었다. 매쉬업 서비스는 다양한 웹서비스가 융합된 서비스를 지칭하며 웹서비스의 인터페이스인 Open API를 통해 보다 손쉽게 지원되고 있다. 하지만 현재의 매쉬업 서비스는 개발자 중심의 서비스 제공 방식으로 인해 개인 맞춤형된 서비스를 제공 받고자 하는 다양한 사용자의 욕구를 충족시키지 못하는 문제점이 있다. 이러한 문제점 해결을 위해 매쉬업을 손쉽게 하는 매쉬업 프레임워크에 대한 연구가 다양하게 이루어지고 있지만 아직 그 수준이 미비하다. 본 논문에서는 동적으로 웹서비스를 합성하여 매쉬업 서비스를 제공하는 매쉬업 프레임워크의 요소기술로써 Open API 자동 합성을 위한 Open API 선별 기법 및 합성 방안에 대해 소개 하였다.

**Key Words** : Open API, REST, SOAP, Automatic Mash-up, API Classification

### ABSTRACT

With the development of web 2.0 and the rapid diffusion of smart devices, current web service, which only provides very limited information and service, has been transformed into user-friendly and comprehensive web service called mash-up service. Mash-up service is defined as comprehensive service that is created by combining variety of web services. When composing a mesh-up service, Open API is used as web service interface. Current mash-up service, however, has fundamental problem in that it could not satisfy the various needs of users who want I-centric personalized service because it is developed and provided by service developers. To overcome these obstacles, a lot of studies on mesh-up framework are being performed for improving mash-up proess though, it is still in beginning stage. In this paper, we introduce an API selection and compostion method as the key technology for mesh-up framework that support the automatic creation of mesh-up service.

### I. 서 론

최근 스마트 디바이스의 보급으로 인해 사용자는

자신에게 맞춤형된 서비스를 제공받고자 하는 욕구가 증대되었다. 이러한 웹 2.0의 발전과 스마트 디바이스의 보급은 사용자들이 현재 제공받고 있는

※ 이 논문은 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구 사업 지원을 받아 수행된 것임(2011-0025-226)

• 주저자 : 광운대학교 전자통신공학과, rlatkd234@kw.ac.kr, 학생회원

o 교신저자 : 광운대학교 전자통신공학과, hwkim@kw.ac.kr, 중신회원

논문번호 : KICS2012-10-494, 접수일자 : 2012년 10월 15일, 최종논문접수일자 : 2013년 6월 17일

단순한 웹 서비스가 아닌 다양한 웹 서비스가 융합된 형태의 매쉬업 (Mash-up) 서비스를 탄생시키는 요인이 되었다. 매쉬업 서비스의 특징은 서비스를 만드는 개발자의 목적에 맞게 기존의 여러 웹 서비스들을 융합해 사용한다는 것이고, 이러한 서비스는 웹 서비스의 인터페이스인 Open API를 통해 보다 손쉽게 지원될 수 있다. 매쉬업 서비스는 주로 Open API들을 융합하는 경우가 많다. Open API를 이용한 기존 매쉬업의 경우에는 여러 Open API들의 Input/Output 값 및 Open API의 모든 정보를 개발자가 각각 같은 형태로 정의해주어야 하기 때문에 일반 사용자의 경우에는 매쉬업 자체에 어려움을 느끼게 된다. 결국 이러한 여러 문제점들은 Open API와 매쉬업 서비스에 대한 활용도를 떨어뜨리고 매쉬업 서비스에 대한 개발이 개발자에게만 국한되어 사용자는 단지 수동적으로 서비스를 제공받는 입장이 되게 한다. 더불어 이러한 문제점은 개인 맞춤형 서비스를 제공 받고자하는 다양한 사용자의 욕구를 막는 장벽이 된다. 따라서 본 논문에서는 사용자에게 필요한 정보를 제공하는 Open API를 직접 선택 할 수 있는 기회를 제공하여, 활용도 및 만족도 높은 서비스를 제공하기 위한 매쉬업 프레임워크의 요소기술 중 기존 매쉬업이 갖고 있는 문제를 해결하고자 Open API 분류기법 및 합성 알고리즘에 대해 연구하였다.

본 논문의 구성은 다음과 같다. 1장 서론에서는 논문의 간략한 Preview를 하며, 2장 본론에서는 Open API와 Mash-up에 대한 간략한 소개와 REST, SOAP 프로토콜의 대해 살펴본다. 3장에서는 본 논문에서 제안하는 Open API 분류 기법 및 합성 방안에 대해 서술 하며, 마지막으로 4장에서 결론으로 마무리 한다.

## II. 관련연구

### 2.1. Open API

Open API는 응용 서비스계층과 통신망의 전달망 계층 사이의 표준화된 인터페이스를 의미하며, 통신망의 기능들을 추상화 시킨 인터페이스로 정의하고 있다. 이를 이용하여 통신망의 구조 및 기술에 있어 독립적으로 새로운 응용 서비스를 쉽게 개발 할 수 있도록 지원하고 있다. 최근 Open API가 통신망에 적용되면서 인터넷 기반의 다양한 응용 서비스들이 등장하고 있으며, 다양한 응용 서비스들은 스마트폰이 보급됨에 따라 모바일 영역으로 까지 확장되고

있는 추세이다. 대표적으로 Open API정보를 포털형식으로 제공하는 외국의 Programmableweb 사이트에서는 지금까지 약 7,100개의 Open API에 대한 정보를 제공하고 있으며 이는 불과 몇 년 전 데이터와 비교 했을 때 배 이상이 증가함을 알 수 있다<sup>[1-3]</sup>.

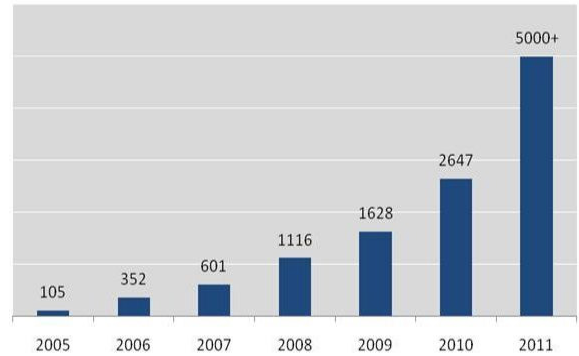


그림 1. 연도별 Open API의 증가 그래프  
Fig. 1. Yearly Increase Graph of Open APIs

### 2.2. SOAP(Simple Object Access Protocol)

SOAP은 ‘Simple Object Access Protocol’의 약어로 HTTP, HTTPS, SMTP등을 사용하여 XML 기반의 메시지를 교환하는 웹서비스의 기본적인 메시지 전달 프로토콜이다. SOAP은 특정 분산 기술에 종속적이지 않고 플랫폼 독립적으로 분산 객체의 리소스에 접근 할 수 있기 때문에 웹서비스와 같은 분산되어 있는 콘텐츠를 서비스 형태로 제공하는 서비스에 쉽게 적용할 수 있으며, 기존의 기술들에 비해 프록시와 방화벽에 구애 받지 않고 쉽게 통신이 가능한 장점이 있다. [그림 2]는 SOAP 프로토콜의 구조를 나타낸다. 모든 SOAP 메시지는 기존의 XML구조와 같이 헤더와 바디를 조합하는 디자인 패턴으로 설계되어 있으며 SOAP envelope, SOAP header, SOAP 본체로 구성되어 있다<sup>[4]</sup>.

### 2.3. REST(REpresentational Safe Transfer)

REST는 월드 와이드 웹처럼 분산 미디어 서비스 시스템을 위한 웹기반의 소프트웨어 아키텍처의 한 형식이다. Roy Fielding의 논문에서 정의된 용어인 REST[5]는 HTTP상에서 SOAP이나 쿠키를 통한 부수적인 레이어나 세션 관리를 추가하지 않고 데이터 전송을 하기위한 간단한 인터페이스로 정의할 수 있으며 구성 요소들 간의 엄격한 분리를 통해 간단한 구현과 확장성을 제공하여 높은 성능을 제공한다.

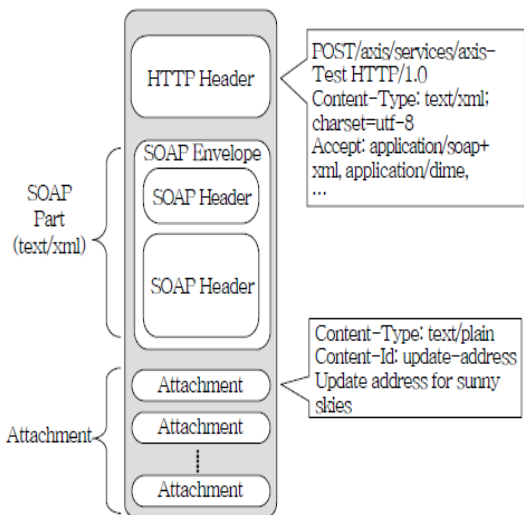


그림 2. SOAP 프로토콜의 구조  
Fig. 2. Structure of the SOAP Protocol

REST구조에서 사용되는 각각의 리소스들은 고유한 URI를 기반으로 동작하며 GET, PUT, DELETE, POST와 같은 HTTP프로토콜을 이용해 정보를 전송한다. 또한 REST에서 사용된 리소스들의 출력 값은 XML, JSON, HTML ,atom과 같은 다양한 방식으로 표현되며 출력 값들을 쉽게 가공하여 사용할 수 있는 장점이 있다

현재 Open API기반의 웹서비스는 SOAP 기반의 서비스에서 REST기반의 서비스로 변화하고 있다. 그 이유는 Open API기반의 웹서비스를 제공하는 업체들이 SOAP 기반의 자율적이지 못하고 단순화되지 않는 서비스 제공방식보다 단순하고 구현이 용이한 REST기반의 서비스를 제공하기 때문이다. 따라서 본 논문에서는 현재 대표적으로 사용되고 있는 REST 기반의 Open API를 대상으로 연구를 진행한다<sup>[6-8]</sup>.

### 2.4. 매쉬업

IT 분야에서는 웹상에서 웹서비스 업체들이 제공하는 다양한 정보와 서비스를 혼합하여 새로운 서비스를 개발하는 것을 의미한다. 이와 같은 매쉬업에는 앞서 살펴본 Open API를 이용한 매쉬업이 대표적으로 사용되고 있으며, 서비스의 대표적인 예로 부동산 정보와 구글맵을 이용한 구글맵 기반의 크레이그 리스트의 부동산 정보 제공 서비스가 있다. 매쉬업은 스마트폰의 단말의 폭발적인 보급으로 인해 수요와 공급 역시 폭발적으로 증가하고 있으며 Open API와 매쉬업 관련 포털사이트인

APIs, mashups and the Web as platform. Learn more »



그림 3. Programmableweb에 등록된 Mash-up 서비스 수  
Fig. 3. Number of Mashup Services registered on Programmableweb Portal Site

Programmable 웹에 따르면 최근 6,758 개의 매쉬업 서비스가 제공되고 있다.

#### 2.4.1. 기존 매쉬업 방안

매쉬업 서비스는 크게 데이터를 제공하는 Data API와 인터페이스를 제공하는 Interface API가 융합되어 생성되며 각 생성된 결과는 웹 페이지를 통해 확인한다. 기존의 매쉬업 서비스는 WMSL(Web Mashup Scripting Language)과 같은 매쉬업 스크립팅 언어로 작성되며, WMSL은 HTML 또는 각 API의 일부분이 코드를 결합하여 GUI를 기반으로 손쉽게 매쉬업 할 수 있도록 도와준다. 하지만 기존의 매쉬업은 서비스 제작에 대한 지식이 부족한 사용자들에게는 도움이 되지 못하며, 자동화된 매쉬업을 제공하지 않아 사용자가 직접 Open API특성을 파악하고 제작해야 하는 문제점이 존재한다.

#### 2.4.2. 수동 매쉬업과 자동 매쉬업

현재 사용자가 쉽고 빠르게 매쉬업을 개발 할 수 있도록 돕는 다양한 기술들을 이용한 매쉬업 도구들이 개발되고 있다. 매쉬업은 크게 개발자가 매쉬업에 사용되는 Open API를 분석하고 매쉬업을 설계 하여 수동으로 매쉬업을 제작하는 매쉬업 언어와 사용자의 요청에 의해 동적으로 매쉬업을 하여 제공하는 동적 매쉬업 도구가 있다. 즉, 개발자가 직접 설계하고 합성하는 수동 매쉬업과 개발자의 관여 없이 사용자의 요청에 따라 자동으로 합성하고 관리하는 동적 합성으로 분류 할 수 있다. 수동 매쉬업에 사용되는 언어는 대표적으로 WSCI(Web Service Choreography InterFace)[9]가 있으며 WSCI는 다른 웹 서비스와 상호 작용을 통한 메시지 유통을 기술하는 XML기반의 인터페이스 기술 언어로 정의 된다. 수동 매쉬업은 개발자가 모든 부분을 수동으로 설계해야 하는 단점이 있으며 배포 후에도 사용자에게 요청에 100% 부합하는 서비스를 제공할 수 없는 문제점이 있다. 동적 매쉬업의 대표적인 시스템으로는 eFlow[10]와 sMash[11]가 있다. 동적 매쉬업 시스템은 복잡한 상호작용을 갖는

Open API들 간의 합성이 불가능한 문제점과 데이터를 제공하는 Open API와 인터페이스를 제공하는 Open API간의 합성 순서를 고려하지 않는 문제점이 있다.

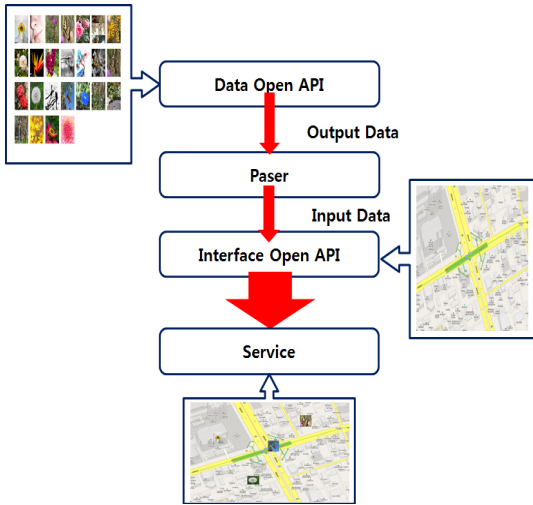


그림 4. 기존의 매쉬업 서비스 생성 방법  
Fig. 4. Current Composition Method for Mash-up Service

따라서 본 논문에서는 기존의 개발자 위주의 수동적인 매쉬업을 벗어나 일반 사용자 중심의 자동화된 매쉬업을 가능하기 위해 기존의 동적 매쉬업 시스템에서 고려되지 않는 Open API 선별방안, Open API 합성방안을 Open API의 출력값 타입과 파라미터 카테고리 이용하여 기술하였다.

### III. Open API 자동 합성을 위한 Open API 선별 방안 및 합성 방안

#### 3.1. Open API 선별 방안

본 절에서는 Open API 자동 합성을 위한 Open API 선별 알고리즘을 제안한다. 매쉬업 서비스는 Data를 제공하는 Data API와 인터페이스를 제공하는 인터페이스 API 또는 새로운 정보를 가공하여 제공하는 또 다른 Data API 나누어진다. 예를들어 지도상에 상품 정보를 제공하는 매쉬업 서비스의 경우 지도를 제공하는 Open API와 상품 정보를 제공하는 Open API가 융합되어 만들어 진다. 이와 같은 경우 상품 정보를 제공하는 Open API는 좌표 정보와 좌표에 해당하는 상품 정보를 지도 Open API에 넘기게 되고 지도 Open API에서 해당하는 좌표에 마커를 찍고 정보를 표시하는 형태로 구현된다. 따라서 서로 다른 Open API를 합성하여 매

쉬업 서비스를 제작하는 경우, 데이터를 제공하는 Open API의 output 데이터는 인터페이스 기능을 제공하거나 새로운 정보를 가공하여 제공하는 Open API의 input 데이터로 사용하지 않으면 자동으로 매쉬업을 할 수 없는 문제점이 있다. 관련연구에서 언급한 자동 매쉬업의 경우 역시 이 부분에 대한 고려가 되어 있지 않다. 따라서 본 논문에서는 자동으로 Open API를 합성하기 위해 데이터를 제공하는 Open API를 Base Open API로 인터페이스 기능을 제공하거나 새로운 정보를 가공하여 제공하는 Open API를 Main Open API로 정의한다. 또한 본 논문에서의 Open API 선별은 Data API기능과 Interface API기능을 통한 Base API, Main API분류를 진행하였으며, 새로운 결과를 제공하는 Data API를 포함한 분류는 추후 논문에 기술할 예정이다.

##### 3.1.1. Base Open API

Base Open API는 서두에서 언급한 바와 같이 다양한 형태의 Open API 중 데이터를 제공하는 Open API를 Base Open API로 정의 한다. 대부분의 Open API가 Base Open API에 해당되며 Base Open API의 정보 제공 분야에 따라 대부분 서비스 카테고리가 정의된다. 대표적인 예로는 다음, 네이버와 같은 곳에서 제공하는 검색 Open API, 트위터, 페이스북과 같은 SNS Open API가 있으며, 현재의 매쉬업 서비스는 이와 같은 Base Open API에 개발자가 인터페이스를 개발하여 제작 되고 있다

Base Open API는 사용자에게 받은 입력 값을 가공하여 사용할 수 있도록 웹에서 사용하는 대표적인 XML, json, atom 형식으로 output 값을 출력 한다.

##### 3.1.2. Main Open API

매쉬업은 데이터를 제공해주는 Open API의 output 값을 받아 서비스에 맞는 인터페이스를 통해 새로운 부가가치 서비스를 만들어 내거나, 또 다른 데이터 제공 Open API의 input값으로 사용하여 새로운 정보를 생산하는 것을 목표로 한다.

따라서 매쉬업에 있어 Main API는 기존의 매쉬업에 사용되는 Interface는 API로 제공되는 것과 사용자가 직접 제작한 인터페이스로 나눌 수 있다. 본 논문에서 기술하는 자동 매쉬업은 기존에 사용되고 있는 Open API간의 자동 매쉬업이기 때문에, Base Open API를 통해 넘어온 output 값을 인터페이스

```

<channel>
  <title>Search Daum Open API</title>
  <description>Daum Open API search result</description>
  <generator>Daum Open API</generator>
  <link>http://dna.daum.net/apis?n11openapi=search</link>
  <lastBuildDate>Fri, 25 May 2012 15:18:16 +0900</lastBuildDate>
  <totalCount>1899546</totalCount>
  <pageCount>796</pageCount>
  <result>10</result>
  <item>
    <author>미디어다음 아고라</author>
    <comment>http://agora.media.daum.net/</comment>
    <title>좌팔 <b>daum</b>, 이명박 풍자사진은 그냥 놔두면서 놓힌 풍자는 삭제</title>
    <link>
      http://bbs2.agora.media.daum.net/gaia/do/kin/read?bbsId=K153&articleId=1322368n11_0pe
    </link>
    <description>
      안녕하세요. <b>Daum</b> 클린센터 입니다. 회원님의 <b>Daum</b> 서비스 이용에 대하여 안내
      확인되어 아래와 같이 조치되었음을 알려드립니다. 경고를 누락되면 로그인이 차단되어 서
    </description>
    <pubDate>20120523115435</pubDate>
    <keyword/>
  </item>
  <item>
    <author>미디어다음 아고라</author>
    <comment>http://agora.media.daum.net/</comment>
    <title><b>Daum</b> 상담사의 열기 언행 (패러디)</title>
    <link>
      http://bbs3.agora.media.daum.net/gaia/do/story/read?bbsId=S102&articleId=5254738n11_c
    </link>
    <description>
      2012년 1월부터 5월까지 << 개그콘서트 >> 보다 더 재미있는 << <b>Daum</b> 상담사와의 대
      실제 있었던 일을 근거로 패러디 한 것입니다. 없었던 사건을 지어내서 장난으로 쓴 것이..
    </description>
    <pubDate>20120520001349</pubDate>
    <keyword/>
  </item>
  
```

그림 5. Open API에 대한 다음 검색 출력 형태  
Fig. 5. Daum Search Output Format for Open APIs

적인 측면과 가공된 정보를 보여줄 수 있는 기능을 제공하는 Open API로 정의 할 수 있다. 하지만 현재 매쉬업에서 인터페이스 및 가공 정보 제공 관련 Open API는 Map 위주의 Open API만 제공되고 있기 때문에 다양한 인터페이스 기능을 제공하는 Main API의 개발이 요구되고 있다.

### 3.1.3. Open API 분류 방안

본 논문에서 제안 하는 Open API분류 방안은 다음과 같다. Open API 정보에는 output 형식에 대한 정의도 함께 분류 하고 있다. 따라서 Base Open API와 Main Open API의 output형식을 이용하여 Base Open API와 Main Open API를 분류 할 수 있다. 앞서 (1),(2)항에서 정의한대로 XML, JSON atom 형식으로 output값을 출력 경우 Base Open API로 정의 하고, 그 외의 경우는 Main Open API로 정의하여 Open API를 분류 하였다.

본 논문에서는 Open API정보를 획득하여 엑셀 파일로 정리 후 자동으로 분류 하는 분류기를 구현 하여 자동으로 분류를 진행하였다. Open API정보는 Open API정보를 제공하는 Programmableweb 사이트에서 가장 많이 사용되는 Open API 정보를 획득 하였으며, 분류된 결과는 Open API 합성에 사용된다.

## 3.2. Open API 합성 방안

본 논문에서는 다양한 매쉬업 서비스들을 분석하

고 분석한 매쉬업 서비스를 통해 기본적인 매쉬업 알고리즘을 도출하여 3.1절에서 제안한 Open API 분류 알고리즘과 연동한 Open API 합성 방안을 기술하고 프로토타입을 구현하였다. 기본적인 매쉬업 알고리즘은 다음과 같다. 사용자가 Data Open API의 쿼리에 원하는 값을 입력하고 해당되는 결과 값을 리턴한 후 파싱을 통해 인터페이스 또는 다른 기능을 하는 Open API의 Input으로 사용하는 형태로 매쉬업이 이루어지고 있다.

앞서 2장 관련연구에서 설명한 [그림 4]와 같은 기존의 매쉬업을 기반으로 자동화된 매쉬업 합성을 진행하기 위해서는 두 가지의 문제점이 발생한다. 먼저 첫 번째 문제는 3.1절에서 설명한 Data Open API와 Interface Open API선별 문제가 있다. 두 번째로는 Open API간의 매쉬업 가능 여부를 판별하는 문제가 있다. 첫 번째 문제는 3.1에서 언급한 Open API 분류 방안으로 해결할 수 있다. 또한 두 번째 문제를 해결하기 위해 기존의 매쉬업 과정을 분석한 결과 매쉬업을 하는 input, output 파라미터 들은 같은 기능을 제공 하는 것을 확인 할 수 있다.

예를 들면 flickr Open API에서 제공하는 위치 정보 파라미터는 구글 맵 Open API의 위치 정보를 관리 하는 파라미터와 매쉬업 되어 해당 사진이 찍힌 장소를 구글 맵에 표시 할 수 있다. 따라서 본 논문에서는 매쉬업이 가능한 같은 기능을 제공하는 파라미터를 그룹화 하여 매쉬업 하고자 하는 Open API를 선택 시 같은 그룹 내에 존재하는 파라미터를 검색하여 해당 파라미터 간의 매쉬업을 진행하는 합성 방안을 제안 하였다.

[그림 6] 은 Open API 합성 시 파라미터 탐색 수행 순서를 flickr와 구글 맵 Open API를 예를 들어 그림으로 표현한 것이다. 사용자가 flickr를 Base Open API로 선택 하였을 때 flickr에 속한 메소드를 탐색한다. 이후 해당 메소드에 속한 파라미터를 탐색한 후 해당 파라미터가 속한 카테고리를 탐색한다. 다음으로 탐색된 파라미터 카테고리에 속한 다른 파라미터들을 탐색하고 앞서 설명한 과정의 역순으로 진행하여 메소드를 탐색하고 API를 탐색한다.

### 3.3. Open API 선별기 및 합성기 구현

본 절에서는 앞에서 기술한 Open API 분류 방안과 합성방안을 구현한다. 시스템 구현을 위한 Open API정보는 Programmableweb 사이트에서 획득하여 사용한다. 또한 Open API 합성 여부 판단

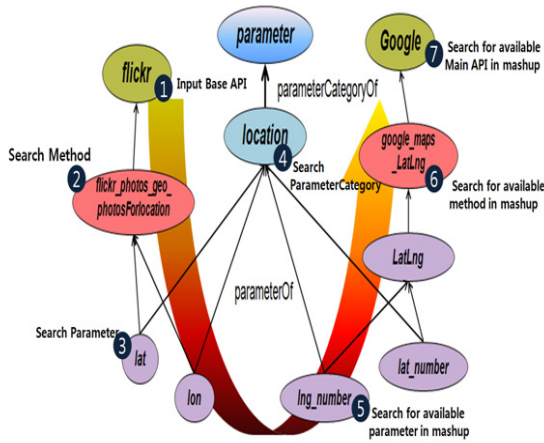


그림 6. 파라미터 탐색 수행 순서  
Fig. 6. Parameter Navigation Sequence

을 위해서 같은 기능을 제공하는 input, output 파라미터를 정리하여 DB로 구축한다. 성능 평가를 위한 시나리오는 다음과 같다. 먼저 매쉬업 하고자하는 Open API를 선택 하고 앞서 구축한 DB를 통해 매쉬업 가능한 Open API와 파라미터를 검색한다. 이후 Open API 분류기가 Base Open API와 Main Open API를 분류하여 파라미터 간의 합성을 진행한다. 파라미터간의 합성은 기존의 매쉬업 알고리즘과 같은 형식으로 선택된 Base Open API의 output 값을 파싱하여 Main Open API의 input으로 사용하여 서비스 합성을 진행한다.

[그림 7]은 자바를 기반으로 구현되었으며 flickr

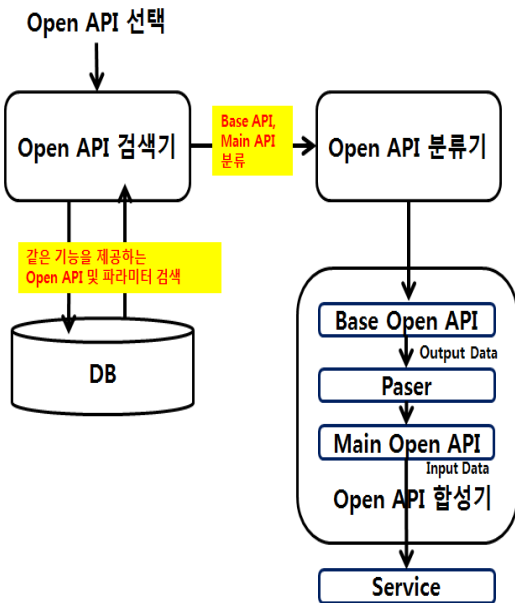


그림 7. Open API 분류기 및 합성기 구조  
Fig. 7. Structure of Open API Classifier and Composer

와 구글 맵 Open API를 사용하여 결과를 출력 하였다.

```

1 API NAME : http://api/Flickr
2 Second Query Result :
http://api.flickr.com/services/rest/?method=flickr.photos.geo.photosForLocation
http://api.flickr.com/services/rest/?method=flickr.photos.geo.getLocation
http://api.flickr.com/services/rest/?method=flickr.photos.search
http://api.flickr.com/services/rest/?method=flickr.people.getInfo
3 3rd Query Result :
http://api.flickr.com/services/rest/?method=flickr.photos.geo.photosForLocation
http://api.flickr.com/services/rest/?lat
http://api.flickr.com/services/rest/?api_key
http://api.flickr.com/services/rest/?method=flickr.photos.geo.getLocation
http://api.flickr.com/services/rest/?photo_id
http://api.flickr.com/services/rest/?method=flickr.photos.search
http://api.flickr.com/services/rest/?text
http://api.flickr.com/services/rest/?max_taken_date
http://api.flickr.com/services/rest/?min_upload_date
http://api.flickr.com/services/rest/?max_upload_date
http://api.flickr.com/services/rest/?min_taken_date
http://api.flickr.com/services/rest/?max_taken_date
4 http://api.flickr.com/services/rest/
4th Query Result :
http://api/location
http://api/location
parameter category Simplification:
http://api/location
5 Mash up > Method URL Method :
http://api/InfoWindow
http://api/LatLng
http://api.flickr.com/services/rest/?method=flickr.photos.geo.photosForLocation
http://api.flickr.com/services/rest/?method=flickr.photos.geo.photosForLocation
    
```

그림 8. 스텝별 Open API 파라미터 검색 결과  
Fig. 8. Stepwise Search Results for Open API Parameters

[그림 8]은 각 스텝 별 Open API 파라미터 검색 결과를 나타낸다. 1번 과정은 사용자가 사용하고자 하는 flickr Open API를 입력하면 2번 과정에서 flickr Open API에 해당하는 메소드를 검색한다. 3번 과정은 각 메소드에 해당하는 파라미터를 검색하고 4번 과정에서는 각 파라미터들이 해당하는 카테고리 검색한다. 5번 과정은 검색된 파라미터 카테고리에 있는 다른 Open API의 파라미터를 검색하고 Base Open API와 Main Open API를 분류

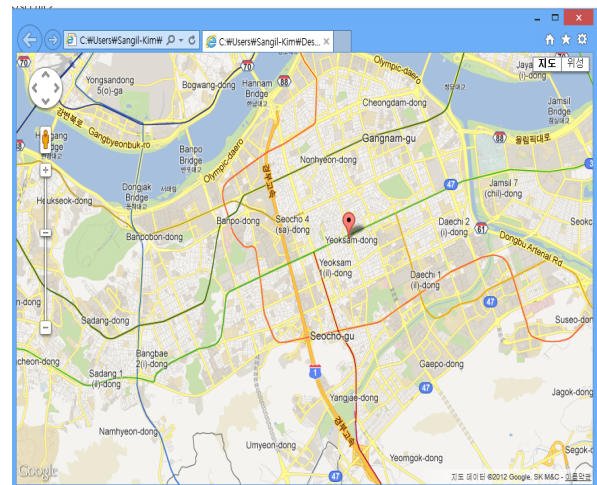


그림 9. 최종적으로 합성된 서비스의 예  
Fig. 9. Example of Finally Composed Service

한다.

[그림 9]는 최종적으로 검색된 Open API 파라미터간의 합성된 그림이다. 앞서 검색한 flickr Open API를 이용하여 사진의 Photo ID를 입력하면 해당 사진의 위치 정보를 output으로 출력하게 된다. 앞서 구현한 Open API합성기에서 해당 output값을 파싱하고 검색된 구글 맵의 마커 파라미터의 input으로 사용을 하면 해당 사진이 찍힌 위치를 구글 지도 상에 마커로 표시 할 수 있다.

#### IV. 결 론

본 논문에서는 기존 매쉬업 방식에 존재 했던 Open API 선별 문제 Open API 매쉬업 가능 판별 문제를 Open API의 정보 및 파라미터 그룹화를 통해 해결 하여 간단한 매쉬업 서비스를 별다른 코딩 없이 생성 하는 것을 보였다. 하지만 Open API간의 매쉬업 가능여부, 많은 수의 Open API파라미터 카테고리 분류 문제를 해결해야할 필요가 있다. Open API 매쉬업 기술은 단말기와 OS, 그리고 단말의 네트워크 자원에 제약 없이 매쉬업 서비스를 제공할 수 있을 뿐만 아니라 사용자의 상황에 따라 맞춤형 서비스를 제작할 수 있게 한다. 따라서 향후 연구로는 여러 Open API를 이용한 다차원적인 서비스 생성이 가능한 Open API에 대한 연구와 위에서 언급한 문제점을 해결한 자동 매쉬업 방안에 대한 연구를 진행할 예정이다. 또한, 사용자 목적에 가까운 서비스를 제공하기 위한 연구도 함께 진행할 예정이다. 본 논문에서는 동적으로 웹서비스를 합성하여 매쉬업 서비스를 제공하는 매쉬업 프레임워크의 요소기술로써 Open API 자동 합성을 위한 Open API 선별 기법 및 합성 방안을 연구하였다.

#### References

[1] M.-S. Kim, S.-I. Kim, J.-M. Kim, J.-H. Ryu, Y.-C. Jung, Y.-M. Park, and H.-S. Kim, "Analysis of foreign target sites for semantic annotation of open API information," *Inform. Soc.*, vol. 17, pp. 117-148, June 2010.

[2] A. K. Moon, Y. M. Park, and S. G. Kim, "Technical trends of semantic annotation for semantic web services," *Electron. Telecommun. Trends*, vol. 25, no. 2, pp.

121-131, 2010, Apr. 2010.

[3] Y.-S. Chang, J.-C. Kim, S.-H. Lee, and Y.-J. Lim, "A study on the mobile mashup development environment for common users," in *Proc. KICS Winter Conf. 2013*, pp. 416-417, Yongpyeong, Korea, Jan. 2013.

[4] W3C Org, *SOAP Messages with Attachments*, December.. 11, 2000, from [Http://www.w3.org/TR/SOAP-attachments.html](http://www.w3.org/TR/SOAP-attachments.html)

[5] Y.-J. Lee and J.-H. Kim, "Semantically enabled data mashups using ontology learning method for web APIs," in *Proc. 2012 Comput. Commun. Applicat. Conf. (ComComAp)*, pp. 304-309, Hong Kong, China, Jan. 2012.

[6] R. Fielding, "Architectural styles and the design of network-based software architectures," Ph.D. dissertation, Dept. Inform. Comput. Sci., University of California, IRVINE, U.S.A., 2000.

[7] W3C Org, *Web Service Description Language (WSDL)*, Retrieved March., 15, 2001, from [Http://www.w3.org/TR/WSDL.html](http://www.w3.org/TR/WSDL.html).

[8] L. Richardson and S. Ruby, *RESTful Web Service*, O'Reilly Media, 2007.

[9] A. Arkin, S. Askary, S. Fordin, W. Jekeli, K. Kawaguchi, D. Orchard, and S. Pogliani, *Web Service Choreography Interface (WSCI) 1.0*, Retrieved Aug., 2002, from <http://www.w3.org/TR/2002/NOTE-wsci-20020808>.

[10] F. Casati, S. Ilnicki, L.-J. Jin, V. Krishnamoorthy, and M.-C. Shan, "Adaptive and dynamic service composition in eFlow," in *Proc. 12<sup>th</sup> Int. Conf. Advanced Inform. Syst. Eng. (CAiSE '00)*, pp. 13-31, Stockholm, Sweden, June 2000.

[11] B. Lu, Z. Wu, Y. Ni, G. Xie, C. Zhou, and H. Chen, "sMash: semantic-based mashup navigation for data API network", in *Proc. 18<sup>th</sup> Int. Conf. World Wide Web (WWW '09)*, pp. 1133-1134, Madrid, Spain, Apr. 2009.

[12] Target Site, *Programmableweb*, [Online],

Available: [Http://www.Programmableweb.com](http://www.Programmableweb.com).

**김 상 일 (Sang-il Kim)**



2010년 2월 서일대학교 정보  
통신공학과 졸업  
2012년 9월 광운대학교 전자  
통신공학과 석사  
2012년 9월~현재 광운대학교  
전자통신공학과 박사과정  
<관심분야> 시맨틱 웹서비스,  
상황 인지 서비스

**김 화 성 (Hwa-sung Kim)**



1981년 2월 고려대학교 전자  
공학과 졸업  
1983년 2월 고려대학교 전자  
공학과(석사)  
1996년 Lehigh Univ. 전산학  
(박사)  
1984년 3월~2000년 2월

ETRI 책임 연구원  
2000년 3월~현재 광운대학교 전자통신공학과 교수  
<관심분야> Wireless Internet, NGN 미들웨어 환경,  
Streaming service