

## 모바일 웹의 동적개발을 위한 설정기법

김철진\*

<sup>1</sup>인하공업전문대학 컴퓨터시스템과

### A Configuration Technique for Dynamic Development of Mobile Web

Chul-Jin Kim<sup>1\*</sup>

<sup>1</sup>Dept. of Computer Systems and Engineering, Inha Technical College

**요 약** 모바일 디바이스의 급속한 보급으로 사용자들은 새로운 모바일 소프트웨어 대한 요구가 증가하고 있으며 소프트웨어 공급자(개발자)는 빠르고 적합한 소프트웨어를 제공할 수 있어야 한다. 그러나 모바일 플랫폼 구조에서는 수정된 모바일 앱을 사용자에게 제공하기 위해서 복잡한 프로세스를 거쳐야 하며 신속하고 즉각적인 업데이트를 제공할 수 없다. 이에 반해 모바일 웹은 다양한 수정 요구사항에 대해 사용자에게 즉각적인 변경을 제공할 수 있기 때문에 모바일 웹의 사용이 증가하고 있다. 따라서 본 논문에서는 모바일 웹 개발 및 유지보수의 생산성을 향상시키기 위한 동적 설정기법을 제안한다.

**Abstract** Owing to the rapid supply of mobile devices, the requirements of users for new mobile software has increased and software vendors (developers) should be able to provide software fast and appropriately. On the other hand, the mobile application in a mobile platform architecture should pass through a complex process to provide the modified mobile application to users and cannot provide an update quickly and immediately. On the other hand, the use of the mobile web is growing because a mobile web can provide an immediate modification of various update requirements to users. Therefore, this paper proposes a dynamic configuration technique for enhancing the productivity of the mobile web development and maintenance in this paper.

**Key Words** : Dynamic Configuration, Mobile Web, Mobile Web Interface, XPDL

#### 1. 서론

소프트웨어는 생명주기(생성→사용→수정→소멸)를 가지며 잦은 변경에 따른 효율적인 관리를 통해 소프트웨어의 생명을 지속적으로 연장시킬 수 있다. 모바일 디바이스의 급속한 보급으로 사용자들은 새로운 모바일 소프트웨어 대한 요구가 증가하고 있으며 소프트웨어 공급자(개발자)는 빠르고 적합한 소프트웨어를 제공할 수 있어야 한다. 현재 모바일 앱은 이러한 모바일 IT 흐름에 대한 한계를 느끼고 있다. 안드로이드 플랫폼 구조에서

는 수정된 모바일 앱을 사용자에게 제공하기 위해서는 복잡한 프로세스를 거쳐야 하며 신속하고 즉각적인 수정을 반영할 수 없다. 모바일 웹은 이러한 사용자의 불편함을 반영하듯 사용이 증가하고 있으며(Fig. 1)[1], 개발 또한 기존 PC 웹 개발 형태와 크게 다르지 않으므로 개발자에게도 많은 부담을 덜어주고 있다. Table 1의 모바일 앱과 모바일 웹 비교에서와 같이 개발 비용, 호환성/이식성, 업데이트 용이성 측면에서 모바일 웹이 장점을 제공하고 있다.

이 논문은 2013학년도 인하공업전문대학 교내연구비지원에 의하여 연구되었음.

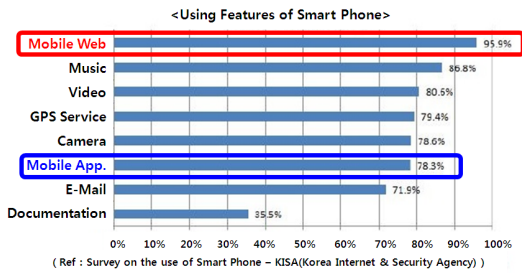
\*Corresponding Author : Chul-Jin Kim(Inha Technical College)

Tel: +82-10-6398-7471 email: cjkim@inhac.ac.kr

Received February 26, 2014

Revised (1st March 18, 2014, 2nd March 24, 2014)

Accepted June 12, 2014



[Fig. 1] Using Trends of Mobile Web and Mobile App.

모바일 웹의 개발은 단순히 웹 페이지 개발을 넘어서 동적인 수정을 통해 즉각적인 변경을 제공할 수 있는 환경이 제공되어야 한다. 또한 개발의 범위를 개발자에서 기획자나 일반 사용자에게도 제공하여 융합된 다양한 서비스가 개발될 수 있는 기반이 마련되어야 할 것이다. 이런 개발 환경을 제공하기 위해서는 동적인 모바일 웹 서비스 개발이 이루어 지야 하며 개발 난이도가 낮은 개발 도구가 절실하다. 본 논문의 동적인 모바일 설정 기법은 이러한 동적인 개발과 개발 용이성을 제공할 수 있는 기반을 제공할 것이다.

[Table 1] Comparison of Mobile Web and Mobile App.

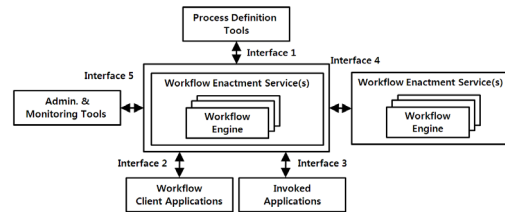
Items	Mobile App.	Mobile Web
Hardware Access	High	Low
Implementation Degree	High	Normal
Running Speed	Fast	Normal
App Store Registration	Possible	Impossible
User Entry	High	Low (Using Browser)
Development Environment	C, Java	HTML, CSS, JavaScript
Development Cost	High	Low
Interoperability	Low	High
Ease of Updating	Low	High

본 논문은 다음과 같이 구성한다. 2장에서는 기존 설정기법과 설정을 위한 커스터마이제이션 관련된 연구를 분석한다. 3장에서는 모바일 웹 동적설정 기법을 위한 구조와 모델을 제안하며, 또한 모바일 웹의 동적 연결구조를 정의한다. 4장에서는 모바일 웹의 동적설정 사례를 통해 본 논문에서 제안한 기법의 적합성을 검증하고, 5장에서 결론과 향후 연구의 방향성을 제시한다.

## 2. 관련연구

### 2.1 WfMC[2]

WfMC(Workflow Management Coalition)는 워크플로우(Workflow) 업체들을 중심으로 구성된 조직으로 워크플로우 관련된 표준들을 정의한다. 일반적으로 WfMC에서 정의하는 표준 구조는 Fig. 2와 같이 프로세스 정의 도구, 워크플로우 엔진, 워크플로우 핸들러 및 UI, 어플리케이션의 4가지 핵심 구성요소들로 구성하며 각각의 구성요소는 5개의 인터페이스들로 구성한다. 인터페이스 1은 정의된 프로세스들 간의 흐름 및 데이터 상호교환을 위한 목적이며, 인터페이스 2는 워크플로우 클라이언트 어플리케이션을 연동하기 위한 인터페이스이다. 인터페이스 3은 워크플로우를 호출하기 위한 인터페이스이며 인터페이스 4는 워크플로우 엔진들 간의 상호 연동을 위한 인터페이스, 그리고 인터페이스 5는 모니터링과 통계 처리를 위한 인터페이스이다.

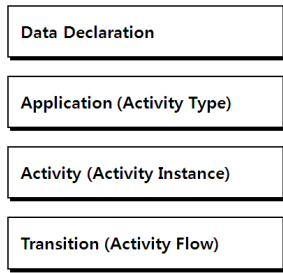


[Fig. 2] Workflow Reference Model of WfMC

WfMC 참조모델에서 프로세스 정의 도구에 사용되는 인터페이스 1인 XPDL(Xml Process Definition Language)[3]이 본 논문에서는 모바일 웹 페이지들 간의 동적인 설정을 위해 가장 적합한 형태로 이용될 수 있을 것이다.

### 2.2 XPDL[3]

XPDL이 프로세스 도구에 활용될 수 있는 구조로서 Fig. 3과 같이 구성된다. 데이터 선언부(Data Declaration)와 프로세스 정의부(Application), 프로세스 인스턴스 정의부(Activity), 그리고 흐름 정의부(Transition)로 구성된다. 이러한 구조는 모바일 웹 페이지 설정 시에도 유사하게 적용될 수 있을 것이다.



[Fig. 3] Structure of XDPL

데이터 선언부는 모바일 웹 페이지들 간에 전달되는 데이터의 타입을 선언할 수 있을 것이며, 프로세스 정의부는 모바일 웹 페이지(또는 모바일 웹 페이지의 인터페이스)의 원형을 정의할 때 이용될 수 있을 것이다. 프로세스 인스턴스 정의부는 모바일 웹 페이지의 인터페이스가 연결될 때 실행 정보들을 정의할 때 이용될 것이다. 마지막으로 Transition 인 흐름 정의부는 모바일 웹 페이지들 간에 동적인 연결 정보를 정의할 때 이용될 수 있을 것이다. 이와 같이 XDPL은 프로세스의 동적인 흐름을 적용하는 것과 유사하게 모바일 웹 페이지의 동적인 흐름을 정의하는데 적합하게 이용될 수 있다. 또한, XDPL은 다른 워크플로우 정의언어 표준과 비교해 봤을 때 <ExtendAttributes>와 같은 추가 태그를 제공하여 더 유연한 기능을 정의할 수 있다.

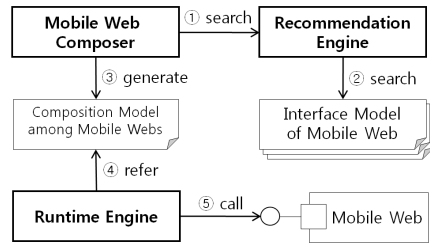
2.3 모바일 앱 설정 기법(안드로이드기반)[4]

안드로이드 플랫폼에서는 인텐트(Intent)와 인텐트 필터(Intent Filter)라는 정보를 이용하여 모바일 앱(Activity) 간에 워크플로우를 설정할 수 있다. 이러한 설정 정보는 XML 형태로 정의되며 개발자에 의해 초기 개발 시 설정할 수 있다. 이렇게 설정된 정보를 모바일 앱으로 패키징(파일 형식 : apk) 되어 사용자의 모바일 디바이스에 설치된다. 이렇게 한번 설치된 모바일 앱은 이러한 설정 파일을 수정할 수 없으며, 수정이 필요한 경우에는 설정 파일을 수정하여 다시 패키징 한 후에 다운로드 받아 재설치해야 한다.

2.4 모바일 웹 추천 기법[5]

연구 [5]는 모바일 웹 간의 효과적인 연결을 위한 추천 기법을 제안한다. 연구 [5]에서는 모바일 웹의 인터페이스에 대한 정의 방법과 이러한 인터페이스를 통해 모바일 웹을 추천하기 위한 기법을 제안한다. 모바일 웹 연결 시

요구 인터페이스에 적합한 모바일 웹을 추천하기 위한 아키텍처는 Fig. 4와 같다.



[Fig. 4] Mobile Web Recommendation Architecture

모바일 웹 조합 시 모바일 웹 조합기(Mobile Web Composer)는 추천 엔진(Recommendation Engine)을 통해 적합한 모바일 웹을 추천받을 수 있다. 모바일 웹 조합을 위해 추천 엔진에 검색(①)을 요구하는 인터페이스 정보를 전달하면 추천 엔진은 요청 인터페이스와 일치하는 인터페이스 정보를 검색(②)하여 가장 유사한 모바일 웹 인터페이스 정보를 추천한다. 추천 엔진에서 모바일 웹의 인터페이스에 일치하거나 유사한 모바일 웹 인터페이스를 제공하기 위해 본 논문에서는 XDPL을 이용하여 제안한다. 모바일 웹 조합기는 추천받은 모바일 웹 인터페이스를 기반으로 조합을 구성하여 조합된 모델정보를 생성(③)한다. 생성된 모델정보는 실행엔진(Runtime Engine)에 의해 참조(④)하여 실제 모바일 웹을 호출(⑤)하며, 실행 시 모바일 웹의 인터페이스는 동적으로 참조되거나 변경될 수 있도록 구성된다.

2.5 모바일 결합을 위한 UI 개발[6]

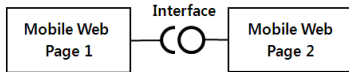
연구 [6]는 서비스 명세 언어인 WADL(Web Application Description Language)[7]을 이용하여 모바일 서비스를 결합하기 위해 UI 코드를 자동생성하기 위한 기법을 제안한다. 연구 [6]에서는 모바일 웹 간의 관계를 정의하기 위한 정형화된 모델을 제안하며 데이터들 간의 흐름을 정의하기 위한 알고리즘을 제안한다. 그러나 연구 [6]은 모바일 웹 서비스를 결합하기 위해 코드를 생성하기 때문에 변경 시 모바일 웹 UI가 다시 생성되어야 한다. 이러한 코드 재생성 과정은 동적 변경을 쉽지 않게 만든다. 따라서 연구 [6]은 모바일 웹의 동적인 개발 및 변경이 미흡하다.

### 3. 모바일 웹 동적설정 기법

모바일 웹 동적설정 기법을 위해 동적설정을 위한 기본 구성요소와 동적설정을 위한 구조, 그리고 XPDL 기반의 동적모델을 정의한다. 또한 XPDL 기반으로 모바일 웹들 간의 연결 구조를 제안한다.

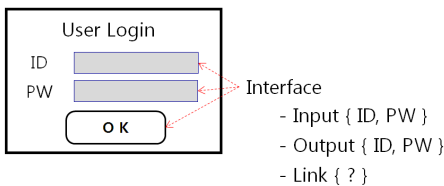
#### 3.1 모바일 웹 동적설정을 위한 구성요소

HTML 기반의 모바일 웹을 동적으로 연결하기 위해서는 인터페이스가 존재해야 하지만, HTML 자체는 텍스트 기반으로 인터페이스가 존재하지 않는다. 따라서 Fig. 5와 같이 모바일 웹 페이지에 인터페이스를 정의하여 동적 설정이 가능하도록 해야 한다.



[Fig. 5] Interface between Mobile Web Pages

모바일 웹에서 인터페이스는 사용자 인터페이스(User Interface)로 보이는 영역을 인터페이스로 정의하면 다른 모바일 웹 페이지와 조합을 할 수 있을 것이다. Fig. 6과 같이 모바일 웹 화면에서 ID를 입력하기 위한 텍스트 필드와 PW를 입력하기 위한 텍스트 필드, 그리고 OK 버튼을 인터페이스로 구성할 수 있다. 이러한 요소들은 입력, 출력, 연결 요소 인터페이스라고 할 수 있다.



[Fig. 6] Interface Case of Mobile Web Page

모바일 웹의 인터페이스를 HTML 코드에서 보면 Fig. 7과 같이 OK 버튼에 대해 <A> 태그 영역 코드가 인터페이스 영역이 된다.

```
<A href = "http://.../LoginCheck.html">OK</A>
```

[Fig. 7] Static Interface Code of Mobile Web Page

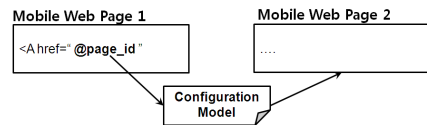
Fig. 7과 같이 인터페이스 코드는 연결되는 페이지가 고정되어 있기 때문에 모바일 웹 페이지 간에 설정이 불가능하며, 동적 변경이 불가능하다. 따라서 Fig. 8과 같이 식별자로 설정하여 동적으로 모바일 웹 페이지를 설정할 수 있도록 인터페이스를 정의해야 한다.

```
<A href = " @LoginCheck_ID ">OK</A>
```

[Fig. 8] Dynamic Interface Code of Mobile Web Page

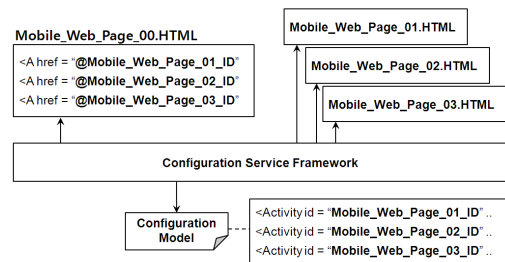
#### 3.2 동적설정 구조

모바일 웹 인터페이스를 통해 동적으로 모바일 웹 페이지를 구성하기 위해서는 Fig. 9와 같이 모바일 웹 페이지들 간에 직접적으로 페이지가 연결되는 것이 아니라 설정 모델(Configuration Model)을 통해 연결될 수 있어야 한다.



[Fig. 9] Concept Model for Connecting Mobile Web Pages

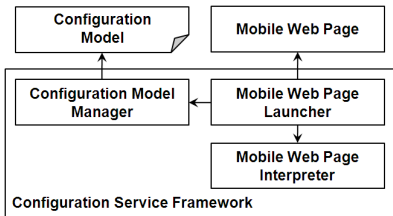
모바일 웹 페이지들 간에 동적으로 연결되기 위한 조건은 각 페이지들 간에 연결 인터페이스를 식별자로 정의해야 하며, 식별자는 설정 모델에 연결되기 위한 모바일 웹 페이지 정보를 포함하고 있다.



[Fig. 10] Configuration Architecture for Connecting Mobile Web Pages

모바일 웹 페이지를 동적으로 연결하기 위한 구조는 Fig. 10과 같다. 설정 서비스 프레임워크(Configuration Service Framework)는 설정 모델에 저장된 식별자들을

통해 실제 모바일 웹 페이지를 찾아서 연결해 준다. 설정 서비스 프레임워크는 설정 모델을 해석하기 위한 기능과 실제 모바일 웹 페이지를 호출할 수 있는 기능을 제공해야 한다.

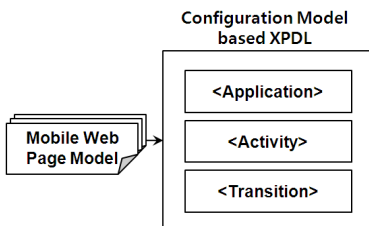


[Fig. 11] Components of Configuration Service Framework

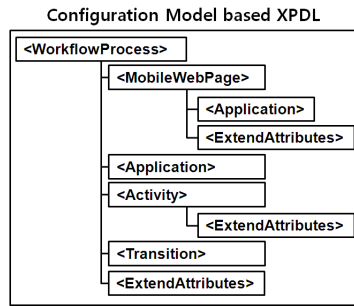
설정 서비스 프레임워크는 Fig. 11과 같이 설정 모델을 해석하기 위한 설정 모델 관리자(Configuration Model Manager)와 모바일 웹 페이지를 호출하기 위한 모바일 웹 페이지 런처(Mobile Web Page Launcher)로 구성된다. 모바일 웹 페이지 해석기(Mobile Web Page Interpreter)는 호출된 모바일 웹 페이지를 해석하여 실행할 수 있다.

### 3.3 동적설정을 위한 모델

모바일 웹 페이지 결합을 위한 동적 설정 모델은 Fig. 12와 같이 XPDL 구성을 따르며 Fig 13과 같이 <WorkflowProcess> 내에 <Application>, <Activity>, <Transition>으로 구성된다. <MobileWebPage>가 하나의 모바일 웹 페이지를 구성하기 위한 단위이다. <Application>과 <Activity>는 모바일 웹 페이지 내의 인터페이스 타입과 인스턴스가 된다.



[Fig. 12] Configuration Model based on XPDL for Dynamic Configuration between Mobile Web Pages(1)

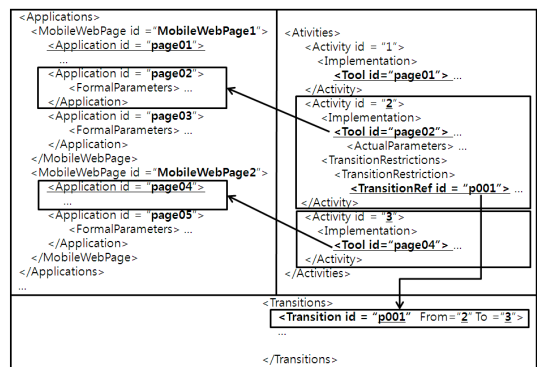


[Fig. 13] Configuration Model based on XPDL for Dynamic Configuration between Mobile Web Pages(2)

<Application>은 모바일 웹 페이지 내의 인터페이스 타입을 정의하기 위한 영역으로 입력 매개변수(<FormalParameters>)와 입력 매개변수의 데이터 타입(<DataFields>)을 정의한다.

<Activity>는 <Application>에서 정의한 모바일 웹 페이지 인터페이스의 인스턴스(Instance)를 정의하기 위한 영역으로 입력 매개변수에 대한 실 데이터(<ActualParameter>)를 정의한다.

<Transition>은 모바일 웹 페이지들 간의 연결을 정의하기 위한 영역으로 호출 모바일 웹 페이지(from 속성)와 호출되는 모바일 웹 페이지(to 속성)를 정의한다. id 속성을 통해 모바일 웹 페이지의 인스턴스인 <Activity>와 연결된다. <Activity> 내의 <TransitionRef>에서 <Transition>의 id를 호출하여 모바일 웹 페이지를 연결할 수 있다.



[Fig. 14] Relation among <Application>, <Activity> and <Transition> for Dynamic Configuration

Fig. 14에서와 같이 <Application>, <Activity>,

<Transition> 간의 관계를 파악할 수 있다. 모바일 웹 페이지 MobileWebPage1은 3개의 인터페이스 page01, page02, page03을 포함하고 있으며, 모바일 웹 페이지 MobileWebPage2는 2개의 인터페이스 page04, page05를 포함하고 있다. 이 5개의 인터페이스 타입에 대해 인스턴스는 <Activity>에 정의하고 있다. <Activity id="2">인 인스턴스는 <Tool id="page02">를 통해 page02 모바일 웹 페이지의 인터페이스 타입을 참조하고 있다. 또한 page02 인터페이스는 <TransitionRef id="pool1">을 통해 <Transition id="p001">을 참조하고 있다. <Transition>은 from과 to 속성을 통해 연결되어야 할 인터페이스를 정의하고 있다. <Transition> 속성 from="2"는 <Activity id="2">를 참조하며 to="3"은 <Activity id="3">을 참조한다. <Activity id="2">은 모바일 웹 페이지 MobileWebPage2의 인터페이스 page4를 참조하고 있다. 따라서 모바일 웹 페이지 MobileWebPage1의 인터페이스 page02와 모바일 웹 페이지 MobileWebPage2의 인터페이스 page4가 동적으로 연결될 수 있다.

```

<TypeDeclarations>
  <TypeDeclaration id = "DataType">
    <ExternalReference location=http://wfmc.org/standards/docs/xpdl_sample/dataTypeSchema.xsd />
  </TypeDeclaration>
</TypeDeclarations>
<WorkflowProcesses>
  <WorkflowProcess id = "flow001">
    <Applications>
      <MobileWebPage id = "MobileWebPage1">
        <Application id = "page01">
          <Application id = "page02">
            <FormalParameters>
              <FormalParameter id = "dataField01" index = "1" mode = "IN">
                <DataType>
                  <BasicType Type="String" />
                </DataType>
              </FormalParameter>
              <FormalParameter id = "dataField02" index = "2" mode = "OUT">
                <DataType>
                  <DeclaredType id="DataType" />
                </DataType>
              </FormalParameter>
            </FormalParameters>
          </Application>
          <Application id = "page03">
            <FormalParameters> ...
          </Application>
        </MobileWebPage>
      </Applications>
    </WorkflowProcess>
  </WorkflowProcesses>

```

[Fig. 15] <FormalParameter> for Input Parameter

모바일 웹 페이지의 인터페이스를 정의하기 위한 <Application>은 입력 매개 변수에 대해 정의할 수 있다. Fig. 15와 같이 <FormalParameters>내에 입력 매개변수의 개수 만큼 <FormalParameter>를 정의하고 데이터 타입을 <DataType>내에 정의한다. 기본 데이터 타입 외에 사용자 정의 클래스 타입을 정의하기 위해서는 <TypeDeclaration>을 정의하여 XML 스키마로 정의하여 참조한다.

```

<Activities>
  <Activity id = "1">
    <Implementation>
      <Tool id="page01"> ...
    </Activity>
  <Activity id = "2">
    <Implementation>
      <Tool id="page02"> ...
      <ActualParameters>
        <ActualParameter>cjkim</ActualParameter>
        <ActualParameter>1234</ActualParameter>
      </ActualParameters>
    </Tool>
  </Implementation>
  <TransitionRestrictions>
    <TransitionRestriction>
      <TransitionRef id = "p001"> ...
    </TransitionRestriction>
  </Activity>
  <Activity id = "3">
    <Implementation>
      <Tool id="page03"> ...
    </Activity>
</Activities>

```

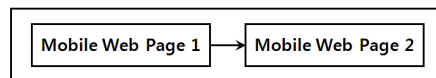
[Fig. 16] <ActualParameter> for Input Parameter

모바일 웹 인터페이스에서 정의한 입력 형식 매개변수에 대해 실 데이터를 정의하기 위해 <Activity> 내에 <ActualParameter>를 이용한다. <Application>의 <FormalParameters>내에서 정의한 입력매개변수의 개수와 데이터 타입에 맞는 <ActualParameter>를 정의해야 한다. Fig. 16에서와 같이 <ActualParameters>는 2개의 문자 타입의 입력 매개변수 실 데이터를 정의하였다.

### 3.4 모바일 웹의 동적 연결구조

모바일 웹 페이지를 동적으로 연결하기 위한 4가지 동적설정 구조를 제안한다. 단순연결, 조건연결, 분기연결, 결합연결로 구성된다. 이러한 연결구조를 설정하기 위해 XPDL의 <Transition> 구조를 이용한다.

단순연결 구조는 Fig. 17과 같이 모바일 웹 페이지들 간에 하이퍼링크(HyberLinK)를 통한 구조를 나타낸다.



[Fig. 17] Simple Connection Structure

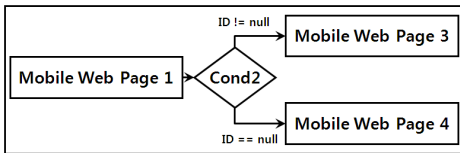
단순연결 구조를 위한 동적설정 코드는 Fig. 18과 같이 <Transition> 의 From과 To 속성에 모바일 웹 페이지의 <Activity> id 값을 설정한다. 단순연결 구조를 동적으로 변경하고자 할 때는 To 속성을 다른 <Activity> 의 id로 변경하면 된다.



```
<Transitions>
  <Transition Id = "p001" From="Mobile Web Page 1"
    To="Mobile Web Page 2">
</Transitions>
```

[Fig. 18] Configuration Code of Simple Connection Structure

조건연결 구조는 Fig. 19와 같이 조건에 따라 서로 다른 모바일 웹 페이지로 연결되기 위한 구조이다. Mobile Web Page 1은 ID 값이 null일 경우 Mobile Web Page 4로 연결되고, null이 아닐 경우 Mobile Web Page 3로 연결된다.



[Fig. 19] Condition Connection Structure

조건연결 구조를 위한 동적설정 코드는 Fig. 20과 같이 <Transition> 내에 <Condition>을 이용하여 조건을 정의하며, 조건값에 따라 Cond2에서 Mobile Web Page 3이나 Mobile Web Page 4로 연결된다. 조건값이나 조건에 의해 연결되는 모바일 웹 페이지의 설정정보를 변경하면 동적으로 조건연결이 변경될 수 있다.

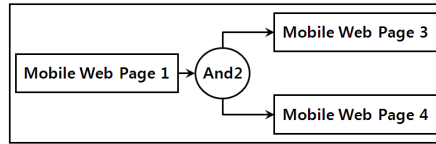
```
<Activity Id="Mobile Web Page 1" />
<Activity Id="Cond2">
  <Split Type="XOR">
    <TransitionRef Id="p002-1" />
    <TransitionRef Id="p002-2" />
  </Split>
</Activity>
<Activity Id="Mobile Web Page 3" />
<Activity Id="Mobile Web Page 4" />

<Transition Id="p001" From="Mobile Web Page 1" To="Cond2"/>
<Transition Id="p002-1" From="Cond2" To="Mobile Web Page 3">
  <Condition>ID != null</Condition>
</Transition>
<Transition Id="p002-2" From="Cond2" To="Mobile Web Page 4">
  <Condition>ID == null</Condition>
</Transition>
```

[Fig. 20] Configuration Code of Condition Connection Structure

분기연결 구조는 Fig. 21과 같이 다른 모바일 웹 페이지들로 분기되어 연결되기 위한 구조이다. Mobile Web Page 1은 Mobile Web Page 3과 Mobile Web Page 4로 분기되어 연결된다. 모바일 웹 페이지에서 이러한 분기

연결 구조는 연결되는 다른 모바일 웹 페이지가 2개 이상의 프레임으로 구성되어 적용되거나, 다른 경우에는 하나의 분기가 화면으로 분기되고 다른 분기는 화면이 없는 기능만 포함된 모바일 웹 페이지로 분기되는 경우가 있을 것이다.



[Fig. 21] Split Connection Structure

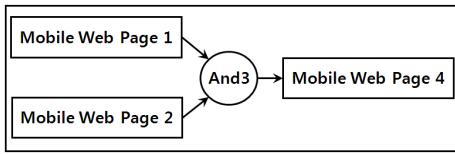
분기연결 구조를 위한 동적설정 코드는 Fig. 22와 같이 분기를 제공하는 <Activity> 내에 <Split> 의 type 속성을 AND로 정의한다. 기존 <Transition>은 기존 단순연결 구조의 정의와 동일하다. 분기연결 되어 있는 모바일 웹 페이지를 동적으로 변경하기 위해서는 <Split> 내의 <TransitionRef>가 참조하고 있는 <Transition>을 변경하면 가능하다.

```
<Activity Id="Mobile Web Page 1" />
<Activity Id="And2">
  <Split Type="AND">
    <TransitionRef Id="p002-1" />
    <TransitionRef Id="p002-2" />
  </Split>
</Activity>
<Activity Id="Mobile Web Page 3" />
<Activity Id="Mobile Web Page 4" />

<Transition Id="p001" From="Mobile Web Page 1" To="And2"/>
<Transition Id="p002-1" From="And2" To="Mobile Web Page 3"/>
<Transition Id="p002-2" From="And2" To="Mobile Web Page 4"/>
```

[Fig. 22] Configuration Code of Split Connection Structure

결합연결 구조는 Fig. 23과 같이 여러 모바일 웹 페이지에서 하나의 모바일 웹 페이지로 연결되는 구조이다. Mobile Web Page 1과 Mobile Web Page 2가 Mobile Web Page 4로 결합되어 연결된다. 모바일 웹 페이지에서 이러한 결합구조는 화면을 제공하는 모바일 웹 페이지와 기능을 제공하는 모바일 웹 페이지가 결합되어 하나의 다른 모바일 웹 페이지로 결합되는 경우이다.



[Fig. 23] Join Connection Structure

결합연결 구조를 위한 동적설정 코드는 Fig. 24와 같이 결합을 제공하는 <Activity> 내에 <Join>의 type 속성을 AND로 정의한다. <Transition>은 기존 단순연결 구조의 정의와 동일하다. 결합연결 되어 있는 모바일 웹 페이지를 동적으로 변경하기 위해서는 <Join> 내의 <TransitionRef>가 참조하고 있는 <Transition>을 변경하면 가능하다.

```

<Activity Id="Mobile Web Page 1" />
<Activity Id="Mobile Web Page 2" />
<Activity Id="And3">
  <Join Type="AND">
    <TransitionRef Id="p003" />
  </Join>
</Activity>
<Activity Id="Mobile Web Page 4" />

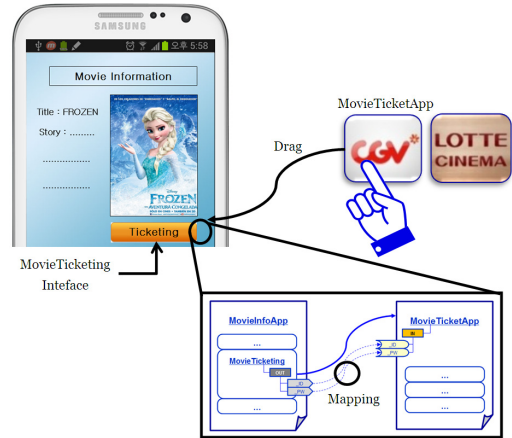
<Transition Id="p001" From="Mobile Web Page 1" To="And3"/>
<Transition Id="p002" From="Mobile Web Page 2" To="And3"/>
<Transition Id="p003" From="And3" To="Mobile Web Page 4"/>
  
```

[Fig. 24] Configuration Code of Join Connection Structure

#### 4. 실험 및 평가

본 연구에서 제안한 동적설정 기법을 검증하기 위해 영화 정보 모바일 웹 페이지와 영화 예매 모바일 웹 페이지를 동적설정으로 연결하기 위한 사례연구를 진행한다. 동적연결 구조 중에 단순연결과 조건연결에 대해 검증한다.

첫 번째 사례연구는 두 모바일 웹 페이지 간에 단순연결을 위한 동적설정을 정의한다. Fig. 25에서와 같이 영화 정보 모바일 웹 페이지의 인터페이스인 예매 버튼을 클릭했을 때 영화 예매 모바일 웹 페이지로 연결되는 구조를 나타낸다. 이 경우 다양한 영화 예매 모바일 웹 페이지가 존재하기 때문에 동적으로 변경하기 위해 동적설정을 정의할 수 있다.



[Fig. 25] Structure for Connecting Movie Information and Ticketing

영화정보 모바일 웹 페이지의 설정 정보를 Fig. 26과 같이 정의할 수 있다. 본 모바일 웹 페이지에는 2개의 인터페이스를 정의하고 있다. <Application>의 id가 MovieInfoApp은 영화정보 모바일 웹 페이지를 실행하기 위한 실행 인터페이스이며 id가 MovieTicketing은 예매 버튼을 나타내는 인터페이스를 나타낸다. 그리고 예매 버튼을 통해 전달되는 데이터는 반환 데이터로 String 타입의 \_ID와 \_PW 2개의 데이터를 전달한다.

```

<WorkflowProcesses>
  <WorkflowProcess id="flow001">
    <Applications>
      <MobileWebPage id="MovieInfoApp">
        <Application id="MovieInfoApp">
          <FormalParameters>...
        </Application>
        <Application id="MovieTicketing">
          <FormalParameters>
            <FormalParameter id="_ID" index="1" mode="OUT">
              <DataType>
                <BasicType Type="String" />
              </DataType>
            </FormalParameter>
            <FormalParameter id="_PW" index="2" mode="OUT">
              <DataType>
                <BasicType Type="String" />
              </DataType>
            </FormalParameter>
          </FormalParameters>
          <Interface for ticketing movie (Ticketing Button)>
        </Application>
      </MobileWebPage>
    </Applications>
  </WorkflowProcess>
</WorkflowProcesses>
  
```

[Fig. 26] Configuration Code for Movie Information

Fig. 27은 영화정보 모바일 웹 페이지와 연결되기 위한 영화 예매 모바일 웹 페이지의 설정정보이다. 현재 영화 예매 모바일 웹 페이지는 CGV 예매를 제공하기 위한 모바일 웹 페이지를 정의한다. 입력 데이터로 2개의 String 타입의 데이터를 정의한다.



```

<MobileWebPage id = "CGV_MovieTicketApp">
  <Application id = "CGV_MovieTicketApp">
    <FormalParameters>
      <FormalParameter id = "_ID" index = "1" mode = "IN">
        <DataType>
          <BasicType Type = "String" />
        </DataType>
      </FormalParameter>
      <FormalParameter id = "_PW" index = "2" mode = "IN">
        <DataType>
          <BasicType Type = "String" />
        </DataType>
      </FormalParameter>
    </FormalParameters>
  </Application>
  <Application id = "...">
    ...
  </Application>
</MobileWebPage>
  
```

Launcher Interface of Mobile Web  
Page "CGV\_MotiveTicketApp"

[Fig. 27] Configuration Code for Movie Ticketing

Fig. 28은 영화정보와 영화예매 모바일 웹 페이지에 대한 인스턴스인 <Activity> 설정정보와 단순연결을 위한 <Transition> 정보를 나타낸다. <Transition>의 id 속성 p001의 From과 To가 <Activity>의 id 1을 참조하고 있는 것은 영화정보 모바일 웹 페이지의 런처(Launcher) 인터페이스를 실행하기 위한 설정정보이다. <Transition>의 id 속성 p002의 From과 To가 2와 3일 참조하고 있는 것은 영화 예매 버튼 인터페이스에서 영화예매 모바일 웹 인터페이스를 연결하는 구조를 정의한다. 이 설정부분을 수정하면 다른 영화예매 모바일 웹 페이지로 동적 변경이 가능하다.

```

<Activities>
  <Activity id = "1">
    <Implementation>
      <Tool id = "MovieInfoApp">
        ...
      </Tool>
    </Implementation>
    <TransitionRestrictions>
      <TransitionRestriction>
        <TransitionRef id = "p001"> ...
      </TransitionRestriction>
    </TransitionRestrictions>
  </Activity>
  <Activity id = "2">
    <Implementation>
      <Tool id = "MovieTicketing">
        <ActualParameters>
          <ActualParameter>_ID</ActualParameter>
          <ActualParameter>_PW</ActualParameter>
        </ActualParameters>
      </Tool>
    </Implementation>
    <TransitionRestrictions>
      <TransitionRestriction>
        <TransitionRef id = "p002"> ...
      </TransitionRestriction>
    </TransitionRestrictions>
  </Activity>
  
```

```

<Activity id = "3">
  <Implementation>
    <Tool id = "CGV_MovieTicketApp">
      <ActualParameters>
        <ActualParameter>_ID</ActualParameter>
        <ActualParameter>_PW</ActualParameter>
      </ActualParameters>
    </Tool>
  </Implementation>
  <TransitionRestrictions>
    <TransitionRestriction>
      <TransitionRef id = "p003"> ...
    </TransitionRestriction>
  </TransitionRestrictions>
</Activity>
  
```

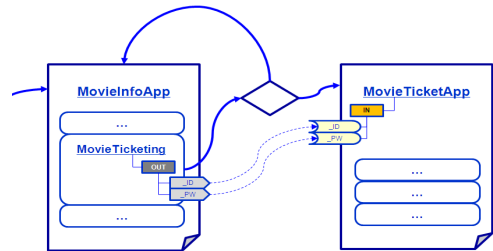
Call Launcher of Mobile  
Web Page Launcher

```

<Transitions>
  <Transition id = "p001" From = "1" To = "1">
  <Transition id = "p002" From = "2" To = "3">
  <Transition id = "p003" From = "3" To = "3">
</Transitions>
  
```

[Fig. 28] Configuration Code for Connecting Movie Information and Ticketing

두 번째 사례연구는 두 모바일 웹 페이지 간에 조건연결을 위한 동적설정을 정의한다. Fig. 29에서와 같이 영화정보 모바일 웹 페이지의 인터페이스인 예매 버튼을 클릭했을 때 조건에 따라 영화 예매 모바일 웹 페이지로 연결되는 구조를 나타낸다. 이 경우에는 영화정보 모바일 웹 페이지에서 입력된 ID와 PW 정보를 체크하여 조건에 따라 영화예매 모바일 웹 페이지로 연결되거나 다시 영화정보 모바일 웹 페이지로 되도록 가도록 정의할 수 있다.



[Fig. 29] Structure for Connecting Movie Information and Ticketing by Condition

Fig. 30은 영화정보와 영화예매 모바일 웹 페이지에 대한 인스턴스인 <Activity> 설정정보와 조건연결을 위한 <Transition> 정보를 나타낸다. <Activity>의 id가 2-1이 영화 예매 버튼을 클릭했을 때 조건을 체크하기 위한 설정정보이다. <Split>에서는 p002-1과 p002-2의 2개 <Transition>을 참조한다. <Transition>의 id 속성 p002-1은 <Condition>내의 ID와 PW가 null이 아닐 경우 <Activity>의 id가 3인 CGV 영화예매 모바일 웹 페이지로 연결된다. 그러나 <Condition>내의 ID와 PW

가 null일 경우 <Activity>의 id가 1인 영화정보 모바일 웹 페이지로 연결된다.

```

<Activity id = "2">
  <Implementation>
    <Tool id="MovieTicketing">
      <ActualParameters>
        <ActualParameter>_ID</ActualParameter>
        <ActualParameter>_PW</ActualParameter>
      </ActualParameters>
    </Tool>
  </Implementation>
  <TransitionRestrictions>
    <TransitionRestriction>
      <TransitionRef id = "p002"> ...
    </TransitionRestriction>
  </TransitionRestrictions>
</Activity>

<Activity id = "3">
  <Implementation>
    <Tool id="CGV_MovieTicketApp">
      <ActualParameters>
        <ActualParameter>_ID</ActualParameter>
        <ActualParameter>_PW</ActualParameter>
      </ActualParameters>
    </Tool>
  </Implementation>
  <TransitionRestrictions>
    <TransitionRestriction>
      <TransitionRef id = "p003"> ...
    </TransitionRestriction>
  </TransitionRestrictions>
</Activity>

<Activity id = "2-1">
  <Implementation>
    <Tool id="Check SSO">
      <ActualParameters>
        <ActualParameter>_ID</ActualParameter>
        <ActualParameter>_PW</ActualParameter>
      </ActualParameters>
    </Tool>
  </Implementation>
  <TransitionRestrictions>
    <TransitionRestriction>
      <Split Type = "XOR" >
        <TransitionRef id = "p002-1" />
        <TransitionRef id = "p002-2" />
      </Split>
    </TransitionRestriction>
  </TransitionRestrictions>
</Activity>

<Transitions>
  <Transition id = "p002" From="2" To ="2-1" />
  <Transition id = "p002-1" From ="2-1" To ="3">
    <Condition> _ID != null and _PW != null </Condition>
  </Transition>
  <Transition id = "p002-2" From ="2-1" To ="1">
    <Condition> _ID == null or _PW == null </Condition>
  </Transition>
  <Transition id = "p003" From="3" To ="3"/>
</Transitions>
    
```

[Fig. 30] Configuration Code for Connecting Movie Information and Ticketing by Condition

본 사례연구를 통해 모바일 웹 페이지를 동적으로 개발할 수 있음을 파악하였으며, 운영 중에 동적설정 정보를 변경하여 동적으로 커스터마이제이션이 가능함을 검증하였다. 이러한 설정 구조 및 설정 모델을 통해 동적인 모바일 웹 개발 도구 개발에 기반을 제공할 수 있다.

### 5. 결론

본 연구에서는 모바일 웹 개발을 위해 동적설정 기법을 XPDL기반으로 제안한다. 또한 동적연결 구조를 제시

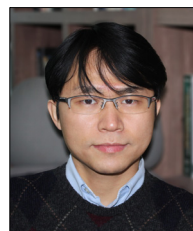
하여 다양한 모바일 웹 연결 구조를 제시하였다. 또한 본 논문에서 제시한 모바일 동적설정 기법을 영화정보와 영화예매 모바일 웹 페이지 연결 사례에 적용하여 동적설정 기법이 적합함을 검증하였다. 향후에는 동적설정 기법을 통해 복잡하고 다양한 모바일 웹 페이지 연결 구조를 개발한다.

### References

- [1] J.M. Lim, etc, "Suvey on the use of Smart Phone", KISA(Korea Internet & Security Agency), 2013.
- [2] Workflow Management Coalition Specification Document, "The Workflow Reference Model.", Version 1.1, November 1994.
- [3] XML Process Definition Language (XPDL) Document Number WFMC-TC-1025: Version 1.14 Document Status, Final October 3, 2005.
- [4] Android Developers, <http://developer.android.com/index.html>
- [5] C.J. Kim and K.S. Choi, "A Mobile Web's Recommendation Technique based on XPDL", Journal of The Korea Academia-Industrial cooperation Society, Vol.14 No.11, Nov, pp.5856~5865, 2013. DOI: <http://dx.doi.org/10.5762/KAIS.2013.14.11.5856>
- [6] Eunjung Lee, "Developing XForms Based Mobile User Interface for Web Service Composition", Korea Information Processing Society, Vol.15D, Issue 6, pp.879-888, 2008.
- [7] java.net, "Web application description language", <https://wabl.dev.java.net/.XUL>, <http://www.mozilla.org/projects/xul/>.

김 철 진(Chul-Jin Kim)

[종신회원]



- 2004년 2월 : 숭실대학교 대학원 컴퓨터학과 (공학박사)
- 2004년 3월 ~ 2009년 2월 : 삼성 전자 책임연구원
- 2009년 3월 ~ 현재 : 인하공전 컴퓨터시스템과 조교수

<관심분야>

컴포넌트 기반 개발 방법론, 컴포넌트 커스터마이제이션, 모바일 서비스, 클라우드 컴퓨팅