

## 이클립스 플랫폼에 기반한 스케일러블 어플리케이션 서비스 저작도구 개발

\*박중빈 \*\*김경원 \*\*\*문재원 \*\*\*\*정종진 \*\*\*\*\*임태범

전자부품연구원

\*jpark@keti.re.kr

Development of Scalable Application Service Authoring Tool  
based on the Eclipse Platform

\*Park, Jong-Bin \*\*Kim, Kyung-Won \*\*\*Moon, Jae-Won \*\*\*\*Jung, Jong-Jin \*\*\*\*\*Lim, Tae-Beom

Korea Electronics Technology Institute

## 요약

스마트폰과 스마트TV와 같은 다양한 콘텐츠 소비 단말들이 급속히 보급되는 최근의 IT환경에서는 콘텐츠 제작, 유통, 소비에 있어서 예전과는 다른 패러다임을 요구하게 되었다. 예를 들어 각기 다른 해상도, 연산량, 사용자환경 등의 다양한 조건들에 대한 고려가 중요한 문제로 부각되고 있다. 이를 해결하고자 단일 서비스 콘텐츠 패키지로 여러 단말들에게 적응적인 서비스를 제공하는 Scalable Application Description Language(SADL)에 기반한 스케일러블 어플리케이션 프레임워크가 제안되었다. 본 논문에서는 이러한 스케일러블 어플리케이션 프레임워크에서 적응형 서비스 콘텐츠를 쉽게 생산하고 패키징 할 수 있도록 이클립스 공개소스 플랫폼 기반의 저작도구를 제안 및 개발하였다. 개발한 저작도구는 유지 보수 및 확장성을 고려하여 모델-뷰-컨트롤러에 기반한 Graphical Editing Framework(GEF)를 이용했다. 제안하는 저작도구는 다양한 운영체제에서 작동하며, 유지보수 및 확장성이 우수하다. 또한 서비스 콘텐츠 개발을 위해 이클립스 플랫폼이 지원하는 다양한 플러그인 도구들을 그대로 활용 가능하다는 장점을 갖는다. 뿐만 아니라 본 저작도구는 기능의 추가적인 확장을 통해 홈 내 가상화 장치들을 위한 응용 프로그램 제작에도 활용할 수 있다.

## 1. 서론

최근 들어 스마트폰, 스마트TV와 같이 다양한 IT단말들이 급속히 보급되고 있다. 특히 사용자들의 개성, 선호도, 경제성 등을 반영하여 단말의 디자인, 해상도, 연산자원 등도 단말의 종류만큼이나 다양해지고 있다. 하지만 단일 해상도 및 특정 기기환경에 최적화되어 개발된 기존의 많은 콘텐츠들은 이러한 다양한 단말 환경에서 그대로 소비되기에는 여러 가지 현실적인 문제가 존재한다. 예를 들어, HD급 해상도에 최적화되어 개발된 어플리케이션 콘텐츠를 QVGA급 해상도를 갖는 장치에서 재생하려고 하면 콘텐츠의 일부 영역만이 표시되거나 영역 스크롤이 발생하는 등의 문제가 발생할 수 있다. 반대로 저사양·저해상도 단말에 최적화된 콘텐츠는 고사양·고해상도 단말에서 소비하기에는 품질 저하 현상이 두드러지게 나타날 수 있다.

이런 문제를 해결하기 위해서는 응용프로그램 및 각종 서비스들이 단말의 특성을 고려해서 사용자환경(User Interface)과 콘텐츠를 적응적으로 제공할 수 있어야 한다. 하지만 이는 단말의 수가 증가함에 따라 여러 버전의 콘텐츠를 유지·관리해야 하는 문제, 자원들의 중복성 문제들이 발생할 수 있다. Scalable Application Description Language(SADL)를 기반으로 하는 스케일러블 어플리케이션 프레임워크는 이러한 문제들을 해결하기 위해 제안된 것이다.[1] SADL은 MPEG-21 DIDL[2]을 기반으로 개발된 마크업 언어의 일종으로써

사용자의 시청 상황, 콘텐츠 소비 단말의 딜리버리 컨텍스트(Delivery Context) 정보를 기반으로 콘텐츠 아이템들을 적응적으로 전송하고 단말기의 환경에 맞춰 서비스 될 수 있도록 설계되었다.[1],[3] 하지만 SADL을 사용하여 적응적 서비스를 제공하기 위해서는 SADL용 콘텐츠를 새롭게 제작해야 한다. 어플리케이션 콘텐츠를 제작하는 입장에서 보면 SADL이라는 새로운 언어를 익혀야 한다는 의미가 될 수도 있는 것이다. 따라서 제작자가 SADL과 관련된 내용을 모르더라도 콘텐츠 제작에 집중할 수 있도록 하는 사용자 친화적인 저작도구의 개발이 수행된 바 있다.[3] 그러나 [3]의 방법에서는 마이크로소프트사의 윈도우즈 운영체제에 특화되어 저작도구가 개발되다 보니 리눅스나 맥 운영체제와 같은 시스템 환경들에서는 직접적으로 설치하여 사용할 수 없는 문제가 존재했다.

따라서 본 논문에서는 저작도구가 다양한 운영체제에서 동작 가능하면서, 개발의 유지보수 능력 및 확장성을 높이기 위해 이클립스[4] 공개소스 개발 플랫폼 기반의 스케일러블 어플리케이션 서비스 저작도구를 개발하였다. 본 논문에서 구현한 저작도구는 이클립스 플랫폼의 다양한 장점들을 그대로 물려받고 있으며, 저작도구와 직·간접적으로 관련되는 다양한 이클립스 플러그인 도구들을 유기적으로 활용할 수 있다는 중요한 장점을 갖는다. 제안하는 저작도구는 XML과 같은 마크업 언어에 익숙한 제작자와 그렇지 않은 제작자를 모두 고려하여 XML에디터와 그래픽 사용자 환경을 동시에 제공한다. 또한 적응형

콘텐츠를 빠르게 제작할 수 있도록 미리 잘 만들어진 예제 프로젝트들을 생성하는 기능도 포함한다. 그밖에도 복사하기/붙여넣기/되돌리기/다시실행 등 편집기가 갖춰야할 필수 기능들을 구현하고 있다.

본 논문은 SADL 언어에 기반한 스케일러블 어플리케이션 서비스 프레임워크 및 이클립스 개발 플랫폼에 대한 간략한 소개와 제안하는 구현방법과 결과를 정리하는 것으로 구성된다.

## 2. 배경 기술

본 절에서는 스케일러블 어플리케이션 서비스 프레임워크를 간략히 설명하고 저작도구가 해당 서비스 플랫폼에서 차지하는 역할을 살펴본다. 또한 이클립스 개발 프레임워크를 간략히 정리하여 구현한 저작도구와 기존의 이클립스와의 관련성을 설명한다.

### 2.1. 스케일러블 어플리케이션 서비스 프레임워크

스케일러블 어플리케이션 서비스를 제공하기 위해서 어플리케이션 콘텐츠는 필수적이며, 이는 SADL 파일을 직접 편집하거나 별도의 저작도구를 이용하여 개발할 수 있다. 저작도구는 SADL언어를 이용하여 다양한 장치들에 대응하도록 각종 콘텐츠 리소스들과 딜리버리 컨텍스트 정보들을 함께 패키징한다.[3] 스케일러블 어플리케이션 서비스는 기본적으로 서버-클라이언트 모델을 따른다.[1],[3] SADL언어로 패키징된 적응형 콘텐츠는 서버에서 클라이언트 단말로 전송되어 처리된다. 이때 서버는 클라이언트 단말로 SADL로 패키징된 전체 정보를 전송할 수도 있으며, 클라이언트 단말의 상황 및 특성에 맞게 SADL로 패키징된 정보 중에서 일부의 리소스와 부가정보만을 전송할 수도 있다. 이렇게 전달받은 적응형 콘텐츠는 클라이언트 단말에서 사용자의 상황에 맞게 구문분석(parsing)을 수행하여 적응적인 사용자 환경을 재현시킬 수 있다.[1],[3]

### 2.2. 이클립스 개발 프레임워크

이클립스는 IBM에서 최초로 개발하여 공개소스 프로젝트로 배포한 개발플랫폼이자 응용프로그램 시장(marketplace)을 갖춘 개발생태계이다.[4]-[6] 이클립스는 통합개발환경(IDE, Integrated development environment) 관점에서 보면 Java를 비롯하여 C, C++, Python, Ruby와 같은 다양한 프로그래밍 언어를 개발하고 시험할 수 있는 도구이다.[4]-[6] 최근의 이클립스는 IDE의 역할을 벗어나 특정 목적을 위한 개발 프레임워크로 인식되고 있는데 이는 이클립스의 모든 구성요소들이 플러그인(혹은 번들)으로 불리는 단위기능 모듈들로 구성되어 있다는 점에 기인한다.[4]-[6] 예를 들어, 만약 현재 사용 중인 이클립스 플랫폼에서 추가로 필요한 기능이 인터넷에 존재한다면 이를 내려받아 자신의 이클립스에 설치 가능하고, 이전에 없는 전혀 새로운 기능의 경우 직접 개발하여 사용할 수 있을 만큼 확장성이 뛰어나다.[4]-[6]

이러한 이클립스 플랫폼에서는 특정 목적의 솔루션 환경을 만들기 위해 크게 2가지 방법을 사용한다. 먼저, 리치클라이언트플랫폼(RCP, Rich Client Platform) 방식으로써, 프로그램 실행에 필요한 핵심 플러그인들을 기반으로 꼭 필요한 요소를 직접 개발해서 붙여나감으로써 독립(standalone)된 프로그램을 만드는 기법이다.[4]-[6] 이는 핵심 기능 위주의 가볍고 빠른 프로그램을 개발해야 할 때 적합하

다. 다른 방법으로는 기존의 이클립스에 새로운 기능을 플러그인 형태로 덧붙이는 방식이다.[4]-[6] 이 방법은 프로그램이 무거워 질 수 있지만 기존의 다양한 기능들을 활용해야 할 때 효과적이다. 본 논문에서는 html파일을 비롯한 다양한 리소스 편집 및 미리보기 기능들이 다수 필요하므로 후자의 개발 방법을 사용했다.

이클립스는 핵심 그래픽 요소를 제외한 대부분의 플러그인들이 다양한 운영체제에서 동작되는 Java 프로그래밍 언어를 이용해서 구현되었다.[4]-[6] 따라서 마이크로소프트 윈도우즈, 리눅스, 맥 운영체제에서 이클립스를 사용할 수 있으며 직접 개발한 플러그인 기능들도 별도의 코드 수정 없이 여러 운영체제 환경에서 동작시킬 수 있다.

## 3. 구현 결과

제안하는 저작도구는 요구사항들을 먼저 정의한 후 이를 달성하기 위해 설계방법을 검토한 후 구현하였다. 본 절에서는 이에 대한 내용 및 최종 구현된 저작도구의 제공기능들을 설명한다.

### 3.1. 저작도구 요구사항

스케일러블 어플리케이션 저작도구는 여러 운영체제 환경에서 동작하는 것이 바람직하다. 또한 프로그램이 수정되는 상황에서도 최종 사용자가 인터넷을 통해 수정된 버전으로 쉽게 갱신할 수 있어야 한다. 기능적 관점에서는, 콘텐츠 제작자들이 도구의 사용법에 시간을 쏟기 보다는 본래의 콘텐츠 제작 업무에 집중할 수 있도록 최대한 복잡한 기술적 요소는 숨기는 과정이 필요하다. 그럼에도 불구하고 숙련된 사용자들을 위해 소스코드나 XML과 같은 상대적으로 복잡한 언어를 직접 편집할 수 있도록 배려하는 것도 필요하다. 또한 최종 단말에서 어떻게 재생될 수 있을지를 검증하기 위한 미리보기 기능이 필요하다.

### 3.2. 저작도구 설계 및 구성

본 논문에서는 상기의 요구조건들을 달성하기 위해 스케일러블 어플리케이션 서비스 저작도구를 위한 특화 기능들을 이클립스 플랫폼에 플러그인 형태로 추가하였다. 먼저 이클립스 상에 저작을 용이하도록 뷰와 에디터들을 적절히 구성·배치하게 하는 저작도구 퍼스펙티브(Perspective)를 구현하였다. 그림 1은 실제 구현되어 실행 중인 스케일러블 어플리케이션 서비스 저작도구를 나타낸다. 그림 1에서는 “KETI Authoring”이라는 퍼스펙티브를 선택하면 이클립스가 제공하는 다양한 기능들 중에서 스케일러블 어플리케이션 서비스 저작에 필요한 핵심 도구들이 로딩되고 화면에 배치된다. 본 저작도구의 작업환경(Workbench)은 “Project Explorer”, “Resource Editor”, “SADL Editor”, “Outline View”, “Properties View”, “Authoring Wizard”와 같은 에디터들과 뷰들로 구성·배치된다. 콘텐츠 제작자는 구현된 저작도구를 이용해서 단위 콘텐츠 리소스들을 제작하고, 이를 SADL파일로 패키징 할 수 있다. 패키징된 정보는 다양한 딜리버리 컨텍스트 환경 하에서 시험되고, 결과를 미리 확인하기 위해 SADL 구문분석기가 포함된 제조함 엔진을 활용하여 이를 수행할 수 있다. 그 밖에도 적응형 콘텐츠를 빠르게 제작하는데 도움이 되는 예제 프로젝트를 생성하는 기능도 포함하고 있다. “Undo”, “Redo”와 같은 편집기가 갖춰야할 기본 기능들도 제공하도록 구현하였다.

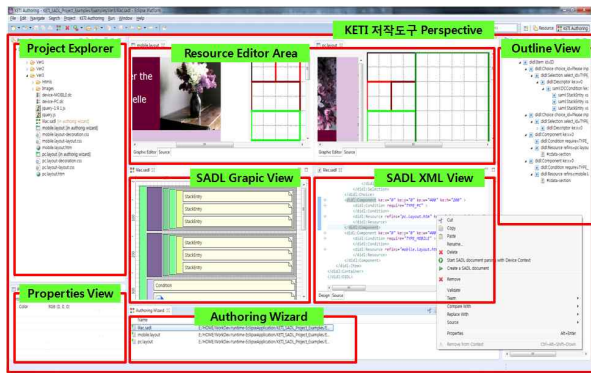


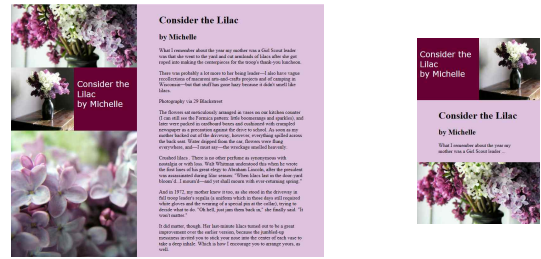
그림 1. 스케일러블 어플리케이션 서비스 저작도구 구성도

세부적인 구현에 사용한 소프트웨어 디자인 패턴은 모델-뷰-컨트롤러(MVC, Model View Controller)방법이다.[7],[8] GEF 프레임워크 [8]는 이러한 MVC 디자인패턴을 이용하여 그래픽 사용자 환경을 필요로 하는 프로그램 개발에 적합하도록 구현한 것으로써 본 논문의 저작도구 개발에도 효과적이다. 디자인패턴의 “모델”에 해당하는 데이터 관리부를 구현하기 위해서는 SADL언어를 구문분석 하는 파서(parser)와 결합하여 자원들의 정보를 관리한다. 실제 사용자와 상호작용하는 “뷰” 부분은 “모델” 정보를 기초로 화면에 “뷰”부분으로 표시하는 정보 전달 경로와, “뷰”를 통해서 사용자로부터 마우스나 키보드를 통해 입력받은 값을 “모델” 정보를 변환하는 정보 전달 경로로 나누어 생각할 수 있다. 이를 처리하기 위해 GEF 프레임워크[8]에서 제공하는 “Model”, “Figure”, “EditPart”, “Editor”, “Factory”라는 개념을 활용하였다.

### 3.3. 저작도구 사용예

제안하는 저작도구를 활용하여 실제 콘텐츠 자원을 SADL로 패키징하고 이를 복원하는 간단한 예시를 살펴보면 다음과 같다. 그림 2는 PC와 Mobile 장치에서 최종적으로 표시되길 원하는 콘텐츠들의 모양이다. 각각은 디바이스별 레이아웃에 사진, 동영상, 텍스트와 같은 기본 콘텐츠 자원(atomic contents resource)들을 배치하여 구성한다. 이런 자원들을 그림 3과 같이 그래픽 에디터를 통해 자원들을 적절히 배치하고, 상황별 조건문을 삽입할 수 있다. 이 결과는 그림 4와 같이 XML파일로 변환된다. 이를 저작도구의 “Authoring Wizard”를 사용하여 패키징하면 간단한 형태의 적응형 서비스를 실현할 수 있다. 이때 딜리버리 컨텍스트 정보를 PC와 Mobile로 각각 설정하고 미리보기를 수행하면 SADL 파일을 편집과 동시에 결과를 미리 확인해볼 수 있다. 본 예시는 매우 간단한 경우이지만 더욱 복잡한 조건들에 대해서도 적용이 가능하다.

제안하는 저작도구는 마이크로소프트 윈도우즈, 리눅스, 맥 OSX 운영체제에서 동작함을 확인했다. 또한 이클립스 3.7 Indigo 버전에서 최초 개발했으나, 이클립스 4.2 Juno, 이클립스 4.3 Kepler 버전에서도 동작됨을 확인했다.



(a) PC용 레이아웃 자원 (b) Mobile용 레이아웃 자원

그림 2. PC와 Mobile 장치를 위한 레이아웃 자원들의 예

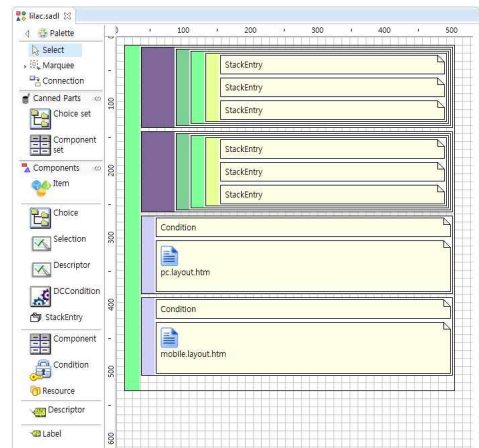


그림 3. 그래픽 환경에서 SADL을 편집하는 예 (Drag & Drop, Copy & Paste 지원)

```
<didl:DIDL xmlns:didl="urn:mpeg:mpeg21:2002:02-DIDL-NS"
xmlns:saml="urn:mpeg:mpeg-m:2010:saml" xmlns:didmodel="urn:mpeg:mpeg21:2002:02-DIDMODEL-NS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:html="http://www.w3.org/1999/xhtml" />
<didl:Container>
<didl:Item id="ID" ke:x="12" ke:y="12" ke:w="493" ke:h="517" >
<didl:Choice choice_id="Please input ID" minSelections="1" maxSelections="1"
ke:x="0" ke:y="0" ke:w="400" ke:h="200" >
<didl:Selection select_id="TYPE_PC" ke:x="0" ke:y="0" ke:w="400" ke:h="200" >
<didl:Descriptor ke:x="0" ke:y="0" ke:w="400" ke:h="200" >
<saml:DCCCondition ke:x="0" ke:y="0" ke:w="400" ke:h="200" >
<saml:StackEntry xsi:type="saml:DSC_Device_Type"
ke:x="0" ke:y="0" ke:w="200" ke:h="200" />
<saml:StackEntry xsi:type="saml:StringValueType" value="PC"
ke:x="0" ke:y="0" ke:w="200" ke:h="200" />
<saml:StackEntry xsi:type="saml:EQ"
ke:x="0" ke:y="0" ke:w="200" ke:h="200" />
</saml:DCCCondition>
</didl:Descriptor>
</didl:Selection>
</didl:Choice>
<didl:Choice choice_id="Please input ID" minSelections="1" maxSelections="1"
ke:x="0" ke:y="0" ke:w="400" ke:h="200" >
<didl:Selection select_id="TYPE_MOBILE" ke:x="0" ke:y="0" ke:w="400" ke:h="200" >
<didl:Descriptor ke:x="0" ke:y="0" ke:w="400" ke:h="200" >
<saml:DCCCondition ke:x="0" ke:y="0" ke:w="400" ke:h="200" >
<saml:StackEntry xsi:type="saml:DSC_Device_Type"
ke:x="0" ke:y="0" ke:w="200" ke:h="200" />
<saml:StackEntry xsi:type="saml:StringValueType" value="MOBILE"
ke:x="0" ke:y="0" ke:w="200" ke:h="200" />
<saml:StackEntry xsi:type="saml:EQ"
ke:x="0" ke:y="0" ke:w="200" ke:h="200" />
</saml:DCCCondition>
</didl:Descriptor>
</didl:Selection>
</didl:Choice>
<didl:Component ke:x="0" ke:y="0" ke:w="400" ke:h="200" >
<didl:Condition require="TYPE_PC" >
<didl:Condition>
<didl:Resource refins="pc.layout.htm" ke:x="0" ke:y="0" ke:w="200" ke:h="200" >
</didl:Resource>
</didl:Component>
<didl:Component ke:x="0" ke:y="0" ke:w="400" ke:h="200" >
<didl:Condition require="TYPE_MOBILE" >
<didl:Condition>
<didl:Resource refins="mobile.layout.htm" ke:x="0" ke:y="0" ke:w="200" ke:h="200" >
</didl:Resource>
</didl:Component>
</didl:Item>
</didl:Container>
</didl:DIDL>
```

그림 4. XML 편집기에서 확인한 SADL 파일

#### 4. 결론

본 논문에서는 스케일러블 어플리케이션 프레임워크에서 적응형 서비스 콘텐츠를 쉽게 생산하고 패키징 할 수 있도록 하는 이클립스 플랫폼 기반의 저작도구를 제안 및 개발하였다. 제안 저작도구는 이클립스 플랫폼의 특성상 다양한 운영체제에서 작동하며, 유지보수 및 확장성이 우수하다. 또한 서비스 콘텐츠 개발을 위해 이클립스 플랫폼이 지원하는 다양한 플러그인 도구들을 그대로 활용 가능하다는 장점을 갖는다.

#### 참고문헌

- [1] Lim, T.-B., Yoon, K., Kim, K.-W., Moon, J.-W., Lee, Y.-J., and Lee, S.-P., "Scalable Application Description Language to Support IPTV Client Device Independence Based on MPEG-21," Web Information Systems and Mining 2011, pp. 361-368, 2011.
- [2] ISO/IEC IS 21000-2 Information technology - Multimedia framework (MPEG-21) - Part 2: Digital Item Declaration, ISO Publication (October 2005).
- [3] 김경원, 문재원, 임태범, 이석필, 배효철, 윤경로, "SADL기반 스케일러블 데이터 서비스를 위한 콘텐츠 저작도구 개발," 2012년도 한국방송공학회 하계 학술대회, pp. 351-354, 2012년 7월.
- [4] <http://www.eclipse.org>
- [5] 에릭 클레이버그, 덴 루벨, *이클립스 실전 플러그인 개발*, 에이콘출판, 2007년.
- [6] 제프 맥아피, 장미셸 르뮤, 크리스 아니슈칙, *이클립스 RCP*, 에이콘출판, 2012년.
- [7] 에릭 프리먼, 엘리자베스 프리먼, 케이스 시에라, 버트 베이츠, *Head First Design Patterns*, 한빛미디어, 2012년.
- [8] <http://www.eclipse.org/gef>