

HTML5 iframe 기반 상호작용형 융합 콘텐츠 저작을 위한 XML 데이터 모형 및 해석기 개발

XML Data Model and Interpreter Development for Authoring Interactive Convergence Contents based on HTML5 iframe

이준정*, 홍준석**, 김우주***

연세대학교 융합기술경영공학과협동과정*, 경기대학교 경영정보학과**, 연세대학교 산업공학과***

Jun Jeong Lee(junjeonglee@yonsei.ac.kr)*, June Seok Hong(junehong@kyonggi.ac.kr)**,
Wooju Kim(wkim@yonsei.ac.kr)***

요약

N-Screen 환경에서 HTML5 표준 기반 콘텐츠 개발은 필연적이다. 그러나 높은 개발 비용과 개발 인프라 부족 등의 이유로 HTML5 조작형 콘텐츠 개발은 소극적이다. 이에 동영상, 오디오와 같은 멀티미디어 콘텐츠를 동적 조작 구현이 가능한 HTML5와 융합하여 사용자와 상호작용이 가능한 효율적인 콘텐츠 개발 모형을 제안하고자 한다. 제안 모형은 통합 제어용 플레이어를 포함한 HTML5 레이아웃 내에 멀티미디어와 iframe(inline frame) 영역을 구분 배치하도록 설계하였다. 상호작용형 HTML5 문서들은 화면 단위로 나누어 작성하여 iframe을 통하여 제공된다. HTML5의 <video>, <audio> 태그를 기반으로 구성된 통합 제어 플레이어는 멀티미디어의 재생 시점과 단위 페이지들이 iframe에 노출되는 시점을 동기화하거나 상호 연동시킨다. 구조적으로 분리된 HTML5 콘텐츠 모형은 개발 난도를 낮출 수 있다. 또한 플래시를 대체하여 HTML5 상호작용 콘텐츠 개발 저변 확대와 관련 시장 활성화에도 기여할 수 있으리라 기대한다.

■ 중심어 : | HTML5 | 상호작용 | 동영상 | 멀티미디어 | 이러닝 |

Abstract

In the N-Screen environment, HTML5 standard-based content development is inevitable. However, it is still passive the development of HTML5 manipulation type contents due to high development cost and insufficient infrastructure. Therefore we propose an efficient contents development model by convergence multimedia contents (such as video and audio) with HTML5 documents that can implement dynamic manipulation for user interaction. The proposed model is designed to divide the multimedia and iframe areas in the HTML5 layout page included the player for integrated contents control. Interactive HTML5 documents are divided into screen units and provided through iframe. The integrated control player composed based on the HTML5 <video> and <audio> tags synchronizes or interlocks the multimedia playback time and the time when unit pages are exposed to the iframe. The structurally divided HTML5 content model can lower the development difficulty. In addition, by replacing Flash, it is expected to contribute to expanding the base of HTML5 interactive contents development and revitalizing related markets.

■ keyword : | HTML5 | Interaction | Video | Multimedia | E-Learning |

I. 서 론

1. 연구의 배경과 목적

1.1 연구의 배경

지난 수년 간 HTML5는 웹 표준 언어로 확대 정착되었고, 플래시(Adobe Flash)의 사용과 지원은 축소되었다. 구글 크롬(chrome)은 2020년 12월 플래시 플레이어의 지원 중단을 발표하였고 이후 웹 환경에서 플래시는 완전히 사라질 것으로 예상된다[1].

다양한 사용자 조작 구성과 유려한 그래픽 퍼포먼스 등 많은 장점에도 불구하고 플러그인(Plugin) 설치로 야기되는 보안 문제는 플래시 퇴출의 큰 원인이 되었다. 지금까지 개발된 플래시 기반 상호작용 콘텐츠들은 조작하는 화면을 녹화한 영상 또는 콘텐츠에 포함된 동영상을 발췌하여 MP4파일 형태로 단순 서비스하거나 폐기되고 있다. 이에 기존 플래시 기반 콘텐츠의 외형을 유지하면서 새로운 디바이스 환경에 대응할 수 있는 범용적인 개발 방법이 필요하다.

스마트기기와 무선 인터넷의 일상화는 이를 통하여 서비스되는 디지털 콘텐츠의 생산, 유통, 소비 전반의 변화를 견인하였다. 특히 스마트폰 등으로도 간편하게 촬영과 편집이 가능해진 동영상 콘텐츠는 강력한 호환성과 글로벌 영상 서비스 플랫폼을 기반으로 생산과 소비가 폭발적으로 증가하였다. 이러한 유형의 동영상들은 대부분 유형에 민감하며 소모적이고 오락성이 강하다.

반면 교육, 설문, 투표, 마케팅과 같은 특정 분야에서는 목적에 부합하는 콘텐츠 제작과 서비스를 위하여 사용자 조작과 같은 직접적인 상호작용이 필요하다. 사용자 조작 과정에서 수집되는 유의미한 사용자 데이터는 다양한 피드백 자료로 활용 가능하다. 그럼에도 HTML5 기술 인력 확보와 서비스 시스템 고도화 등에 따른 시간적, 비용적 문제로 단기간 내에 플래시와 유사 수준의 HTML5 상호작용형 콘텐츠의 개발환경의 활성화는 어렵다.

1.2 연구의 목적

최근 글로벌 동영상 서비스 플랫폼은 거의 무한대의 저장 공간과 다양한 시스템 편의 기능을 제공하면서 많

은 사용자들을 확보하고 있다. 자체적인 수익모델을 통해 자발적이고 적극적인 콘텐츠 유통과 소비를 주도하며 사회 전반에 막대한 영향력을 발휘하고 있다. 그러나 콘텐츠 제작과 서비스 자체가 플랫폼에 종속됨으로써 다양한 콘텐츠 구성과 서비스 요구에 모두 부합하기는 어렵다. 콘텐츠 다양성을 확보하면서도 이러한 제약 상황을 효율적으로 해결하기 위하여 어떤 브라우저에서도 별도의 플러그인 없이 크로스플랫폼 지원이 가능한 HTML5 기반 개발 방법론이 필요하다.

2014년 10월 표준안 발표 이후 HTML5 기반 상호작용 콘텐츠의 개발은 주로 하드 코딩 방식으로 진행되었다. 소스코드에 직접 데이터를 입력하는 하드 코딩 개발 방식은 다양한 오브젝트들을 사용하여 화면상에 독립적으로 디자인과 조작 기능을 표현하기에 적합하다. 그러나 HTML5 문서가 복잡하고 길어지면서 퍼블리싱 작업자의 숙련도에 의해 콘텐츠 품질과 개발 속도가 좌우되는 문제가 따른다. 숙련된 기술 인력의 부족과 예측 불가능한 개발 일정은 HTML5 상호작용 콘텐츠 개발에 있어서 비용 상승의 요인이 되고 있다. 이에 HTML5 문서의 분량을 축소하고 하드 코딩에 포함되는 데이터 연동 부분의 구조적인 분리를 통하여 개발 자원의 효율적 운영을 지원할 수 있다.

본 연구에서는 동영상 또는 사운드와 같은 멀티미디어 콘텐츠와 조작형 HTML5 문서를 융합한 상호작용 콘텐츠의 유형과 구조를 정의하고, 이를 동기화하거나 제어하기 위한 통합 플레이어 기술을 함께 제시한다. 또 콘텐츠 모형 구조와 구현 사례를 xml 표준으로 정리하여 공유함으로써 향후 다양한 방식의 활용을 지원하고 부가 기능을 연동할 수 있는 기초로 삼고자 한다. 특히 영상과 HTML5 문서의 구조적 분리를 통하여 개발 프로세스를 구분하고 개발 난도를 낮춤으로서 HTML5 상호작용 콘텐츠의 개발 편의성을 높이고, 콘텐츠 개발 인프라의 확대와 개발 환경의 발전에 기여할 것으로 기대한다.

2. 콘텐츠 개발 방법 및 연구 범위

본 연구에서는 기존의 상호작용 콘텐츠를 HTML5 기반으로 개발하기 위한 구조를 설계하고 이를 실제 콘텐츠 개발에 적용하였다. 콘텐츠 개발은 동일한 내용과

조작 기능에 대하여 HTML5 단일 문서(Single Document)에 대한 하드 코딩 개발 방식과 HTML5 iframe 분할 구조 방식으로 각각 진행하였다. 이 과정에서 각각의 개발 프로세스와 결과물에 대한 차이점도 함께 살펴보았다. 마지막으로 산출된 결과물에 대하여서는 실무자 대상 심층 면접 조사를 통하여 개발상의 효율성이 있는지를 검토하였다. 연구 과정별 상세 내용은 다음과 같다.

○ 상호작용 콘텐츠 유형의 분석 및 설계

이러닝으로 대표되는 상호작용 콘텐츠에서 멀티미디어와 사용자 조작 기능은 범용적으로 사용되는 콘텐츠 구성 요소이다. 이에 현재 서비스 중인 원격교육용 이러닝 콘텐츠를 기준으로 멀티미디어를 활용한 상호작용 콘텐츠의 일반적인 유형을 선정하였다. 그리고 선정된 유형에 포함된 파일 포맷과 표현방식을 고려하여 콘텐츠 구현에 필요한 개발 구조를 설계하였다. 본 연구에서는 멀티미디어와 조작 기능을 포함한 HTML5 문서를 동일 레이아웃에 하나의 콘텐츠로 융합하여 구성하기 위한 구조적 작성을 중심으로 다룬다. 따라서 멀티미디어의 촬영, 편집, 녹음이나, HTML5 단위 페이지 문서의 소스 코드 작성은 전통적인 방식으로 제작함을 전제한다.

○ HTML5 iframe 방식 콘텐츠 개발

본 연구에서는 HTML5 하드 코딩 개발 방식에서 개발 난도에 많은 영향을 끼치는 콘텐츠 간 동기화와 HTML5 문서 분량 문제를 해결하기 위하여 iframe을 활용한 설계를 고려하였다. HTML5에서 표준 기술로 지원하는 iframe 기능은 멀티미디어 융합형 콘텐츠에서 부모 화면과 독립적인 디자인과 조작 구성이 필요한 하위 HTML5 콘텐츠를 배치하기에 적합하다. 이에 제안하는 모형은 시간값 연동 데이터와 통합 제어 플레이어를 포함한 HTML5 레이아웃 컨테이너 문서를 따로 구성하고, 구성된 레이아웃 컨테이너 안에 멀티미디어와 HTML5 iframe 영역을 구분하여 배치한다. iframe에서는 멀티미디어와 연계된 비선형 정보 또는 조작 기능을 포함한 HTML5 문서가 노출되도록 한다.

본 연구의 콘텐츠 개발 과정에서 도출된 HTML5 콘

텐츠 전체 구조와 동기화를 위한 시간값 데이터 연동 방식은 XML(Extensible Markup Language) DTD(Document Type Definition)와 실제 개발 콘텐츠에 적용된 XML 데이터 문서로 정리하여 공유한다.

또 HTML5 iframe 개발 방식과의 명확한 대조를 위하여 동일 내용의 콘텐츠를 HTML5 하드 코딩 개발 방식으로도 함께 개발한다.

○ 실무 개발자 대상 집단 심층 면접

레이아웃 컨테이너와 iframe으로 보이는 세부 HTML5 단위 페이지 콘텐츠는 구조적으로 분리되어 독립적인 개발이 가능하다. 개발 유형이나 난이도에 따라 기술 인력을 차별적으로 운영할 수도 있다. 이에 기존 개발방식과 동일하거나 유사한 기획, 검수 및 영상 관련 업무는 제외하고 HTML5 퍼블리싱 업무만을 중심으로 기술 인력 등급에 따른 실제 콘텐츠 개발 소요 기간에 대하여 실무자들을 대상으로 심층 면접 조사를 진행하였다. 심층 면접 조사는 HTML5 콘텐츠 개발 전문 업체에 소속된 개발 인력 7인을 대상으로 하였으며, 이들의 업무 경력과 대비하여 콘텐츠 레이아웃 컨테이너를 포함한 프로토타입 개발 소요 일정과 시리즈형 양산 콘텐츠 개발 소요 일정을 각각 파악하여 효율적인 인력 운영이 가능한지 검증한다.

II. 이론적 고찰

1. HTML5 기술의 이해

1990년대 마이크로소프트의 인터넷 익스플로러(IE)는 출시 이후 10여년 이상을 사실상 웹 브라우저의 표준으로 자리매김 해 왔다. 그러나 표준 코딩과는 무관하게 IE에서만 동작하는 확장코드의 사용 증가와 웹 비표준 기술(ActiveX 등)의 남용은 웹 호환성과 보안을 취약하게 만드는 심각한 문제였다. 다양한 콘텐츠 제작의 필요성과 안전하고 편리한 서비스에 대한 시장 요구에 부응하기 위하여 ActiveX를 대체하면서도 어떤 브라우저에서도 제약 없이 서비스가 가능한 웹 표준 문서의 확장은 필연적이었다[2].

기존 HTML 개발환경에서는 비디오와 같은 멀티미

디어나 사용자 조작을 위한 확장 기능을 지원하는 웹 표준이 존재하지 않았다.

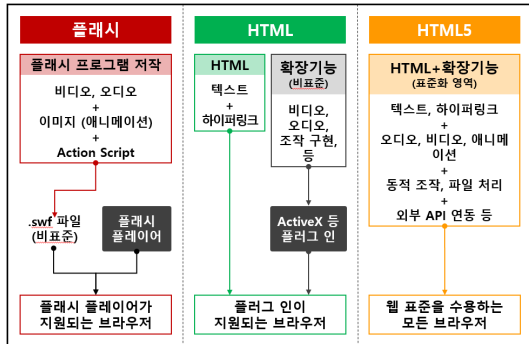


그림 1. 플래시, HTML, HTML5의 기능 및 구현 비교[3]

[그림 1]은 플래시, HTML, HTML5에 대한 각각의 기능적 특징과 브라우저 서비스 방식을 비교한 도식이다. 플래시는 다양한 화면 표현과 역동적인 애니메이션 효과는 물론 액션스크립트를 이용한 사용자 조작 기능을 자유롭게 활용하여 콘텐츠 제작이 가능하였다. 그러나 플래시 프로그램의 저작 산출물인 .swf 파일은 전용 플래시 플레이어가 설치되지 않은 브라우저에서는 구동할 수 없었다. HTML은 단순한 화면 마킹을 위한 텍스트와 하이퍼링크 중심 언어로 이러한 기능을 구현하기에는 한계가 있었다. HTML을 이용하여 다양한 형태의 콘텐츠를 구성하기 위해서는 별도의 프로그램을 설치하고, 브라우저 별로 필요한 확장코드를 지원하는 것이 필요했다. HTML5는 이러한 문제들을 해결할 수 있는 진화된 웹 프로그래밍 언어로서 오디오, 비디오, 그래픽은 물론 동적 조작과 다양한 외부 기능 연동까지 웹 자체에서 처리할 수 있는 웹 표준이다[3].

[표 1]은 HTML5에서 새롭게 지원되는 주요 기술과 그 특징을 정리한 것이다. HTML5의 다양한 기능과 외부 API 연동 기능 등을 이용하여 웹 어플리케이션 개발이 가능하게 되었으며, 다양한 사용자 경험도 제공할 수 있게 되었다. 별도 플러그인 없이도 비디오와 오디오의 재생이 가능하여 멀티미디어를 활용한 풍부한 콘텐츠 제작과 서비스도 가능해졌다. HTML5는 웹 표준 규격을 준수하는 상호작용 콘텐츠를 제작하기에 매우 효율적인 저작 도구이다. 최근에는 네이티브 앱(Native

App)을 대신할 수 있는 웹 앱의 프로그래밍적인 요구가 커지면서 CSS(Cascading Style Sheet)와 자바스크립트(JS), 표준 API를 전문적으로 활용하는 프론트엔드(front-end) 개발에 대한 요구가 더욱 확장되고 있다[4][5].

표 1. HTML5에서 개선된 주요 기술 및 특징[6]

구분	HTML5 기술 특성	속성
Semantics	구조화되고 다양한 기능을 포함한 HTML 태그 제공	풍부한 웹 문서 표현 가능
Multimedia	비디오, 오디오 자원 가능 자체 멀티미디어 가능 포함	ActiveX와 같은 외부 플러그인 제거
Offline & Storage	Offline 환경에서도 웹 이용이 가능한 로컬 스토리지, DB, File 액세스 처리 가능	네트워크 단절 시 연속 처리 및 데이터 저장 문제 해결 (오프라인 환경에서 콘텐츠 사용 가능)
3D, Graphics, Effects	SVG, 캔버스, WebGL 등을 통한 다양한 2차원/3차원 그래픽 기능의 제공	외부 플러그인 없이 2D/3D 그래픽 처리 (플래시 수준의 모션 효과 적용)
Device Access	GPS, 카메라, 동작 센서 등 디바이스의 하드웨어 기능을 웹에서 직접 제어	웹 기반의 디바이스 제어 기능을 통한 웹 애플리케이션 개발 가능
Performance & Integration	비동기 통신, 다중 스레드 기능 등을 통한 웹 처리 성능 향상	웹 성능 문제 개선
Connectivity	클라이언트와 서버 간의 효율적인 통신 기능 제공으로 웹 기반 커뮤니케이션 효율 강화	웹에서의 다양한 통신 기능 제공 (메시징, 응용간 통신 등)
CSS3 Styling Effect	기존 웹 문서의 변경과 성능저하 없이 웹 어플리케이션의 UI (스타일과 효과 등) 기능 강화	N-스크린 서비스 제공 가능

2. 상호작용 콘텐츠

상호작용(interaction)은 '쌍방향'을 뜻하는 Inter와 '활동, 작용'을 뜻하는 Action의 합성어로서 대체로 두 개 이상의 요인이 조합되어 결과에 영향을 미치는 작용을 의미한다. 교수-학습 상황에서는 사용자의 능동적인 참여와 활발한 인지 작용을 끌어내는 쌍방향 활동으로 정의한다.

최근의 콘텐츠 상호작용 개념은 사용자의 조작 경험을 극대화한 형태로 진화하고 있다. VR, AR 또는 홀로그램과 같은 가상현실 기반의 체감형 상호작용 콘텐츠와 인공지능 스피커나 가상 캐릭터 또는 챗봇과 같은 대화형 상호작용 콘텐츠 개발도 활발히 진행되고 있다.

[표 2]는 이러닝 콘텐츠의 사용자 중심으로 상호작용 유형을 구분한 것이다. 상호작용 유형 중 사용자와 콘텐츠 간의 상호작용과 사용자와 시스템 간의 상호작용

은 콘텐츠에 포함된 동적 조작 기능과 연관된다.

표 2. 이러닝의 상호작용 유형기

상호작용 주제	상호작용 예
사용자-튜터	지식과 정보 전달, 학습 목표 제시 및 안내, 학습동기 제공 및 학습 피드백의 제공 등
사용자-사용자	사용자간 메시지 교환 및 정보의 교환, 학습자 간 상호 교수를 통한 학습 동기 고취 등
사용자-콘텐츠 (또는 매체)	콘텐츠 내용의 직접 조작, 퀴즈 및 평가의 수행, 게임 및 과제 실행 등.
사용자-시스템	조작 결과 피드백, 평가 피드백, 성취에 따른 보상, 사용 이력 관리 및 현재 상태 파악 등

멀티미디어 플레이어, 퀴즈 및 정오답 확인, 드래그 앤 드랍, 클릭 또는 터치, 텍스트 입력, 선 그리기, 색칠하기 등과 같은 사용자 조작과 이와 연계된 다양한 활동과 게임 등이 활용된다. 사용자 조작 결과에 따라 콘텐츠의 변경, 다음 학습의 연결, 보상 피드백 등은 시스템 상호작용으로 진행될 수 있다.

본 연구에서는 일반적인 이러닝 콘텐츠의 조작 기능을 참고하여 유사 수준의 HTML5 상호작용 콘텐츠를 효율적으로 개발하기 위한 구조적 모형을 제시하고자 한다. 제안되는 구조적 모형은 사용자와 콘텐츠, 콘텐츠와 콘텐츠 간에 복합적으로 발생하는 다중 상호작용을 지원할 수 있도록 한다. 이와 함께 사용자의 조작과 동시에 멀티미디어와 HTML5 문서가 실시간으로 동기화되는 방안을 중점적으로 다룰 예정이다. 따라서 HTML5 단위 페이지의 기능 구현을 위한 코딩 관련 기술은 본 연구 단계에서는 언급하지 않는다.

3. HTML5 개발 인력 인프라 현황

사용자의 적극적인 조작 참여가 필요한 특정 콘텐츠 영역에서는 프론트엔드 기술 활용이 매우 유용하다. 다양한 파일 포맷을 활용한 시각적, 청각적 효과와 동적 조작의 적절한 구성은 사용자와 콘텐츠 간의 직접적인 상호 작용을 지원할 뿐 아니라 이 과정에서 발생한 사용자 데이터의 수집도 가능하다.

상호작용이 가능한 융합 콘텐츠의 개발은 웹 퍼블리셔로 불리는 UI 개발 인력을 통하여 진행된다. 그러나 최근까지도 표준화가 진행 중인 HTML5 퍼블리싱 기술 인력은 턱없이 부족하다. 디자인 요소와 CSS를 이용

한 시각적 표현과 함께 DOM 기반 동적 프로그래밍을 위한 JS, Angular, React, Vue와 같은 프레임워크 등을 아우르는 기술 숙련에는 수년에 걸친 교육 훈련과 실무 적용 기간이 필요하다[8].

전문 인력의 부족은 HTML5 기반 상호작용 콘텐츠 개발 확산을 저해하는 심각한 요인 중 하나이다. 본 연구에서는 HTML5 콘텐츠 개발 주체들이 참고할 수 있는 개발 구조를 고안하여 그 모형을 제시하고자 한다. 또한 콘텐츠 구현에 필요한 기술 난이도에 따라 개발 인력을 분산 운영할 수 있도록 가이드라인을 제시함으로써 개발 활성화를 지원하고자 한다.

4. 디지털 콘텐츠의 개념과 유형의 분류

사전적 의미의 디지털 콘텐츠(digital contents)란 통상 아날로그적 콘텐츠인 문자, 음성, 음향, 이미지, 그리고 영상과 같은 콘텐츠 등의 각종 정보 형태를 0과 1이라는 비트(bit)단위로 디지털화한 콘텐츠를 총칭하는 개념을 말한다[9].

최근에는 온라인상에서 제공되는 콘텐츠와 콘텐츠 간의 결합 또는 융합으로 재생산된 콘텐츠까지 포괄하는 의미로 사용되고 있으며 그 구분이 모호하고 무의미하다. 다만 본 연구에서는 HTML5 콘텐츠 개발에 필요한 대표적인 디지털 콘텐츠를 형태적 측면에서 동영상, 사운드, 텍스트, 이미지로 구분하여 주요 특성을 도출하였다[10].

표 3. 특성에 따른 유형별 디지털 콘텐츠 분류

콘텐츠	특성	주요상호작용	속성
동영상	시계열성, 연속성	시각, 청각	동적, 선형
사운드	시계열성, 연속성	청각	동적, 선형
텍스트	비 연속성, 고착성	시각	정적, 비선형
이미지	비 연속성, 고착성	시각	정적, 비선형

[표 3]은 유형별 콘텐츠의 특성을 정리한 표이다. 동영상과 사운드는 시간의 흐름에 따라 정보가 전달되는 동적·선형 콘텐츠로, 텍스트와 이미지는 정지된 화면에서 정보가 전달되는 정적·비선형 콘텐츠로 나누어 볼 수 있다. 특히, 시간의 흐름에 따라 정보가 동적으로 변화

하는 동영상과 사운드 콘텐츠를 통합하여 멀티미디어 콘텐츠 또는 멀티미디어로 지칭하기도 한다.

전통적인 플래시 기반 콘텐츠는 멀티미디어의 시간적 흐름에 따라 필요한 시점에 이미지, 텍스트와 같은 정적인 정보를 제공하거나 정지된 시점에서 사용자 조작 활동을 유도하여 콘텐츠와 사용자 간의 직접적인 상호작용에 집중하였다. 또 각 시점에 발생하는 상호작용 정보의 수집과 분석을 통하여 CMS(Content Management System), LMS(Learning Management System) 등과 연계된 적절한 피드백과 서비스 계획을 수립할 수 있도록 지원하였다.

본 연구의 콘텐츠 개발 모형은 이미 검증된 상호작용 콘텐츠의 화면 설계 및 구현 방식을 승계하면서 기술적으로는 HTML5 표준을 적용한다. 이 때 구현되는 시각적 효과나 상호작용 퍼포먼스는 기존 형식과 유사하거나 동일한 수준으로 유지할 수 있도록 유의하여 설계한다.

5. 인터랙티브 비디오 vs 하이퍼비디오

좁은 의미의 인터랙티브 비디오는 영상 위에 사용자가 클릭 가능한 객체가 포함된 비디오로 볼 수 있다. 그러나 상호 작용의 측면으로 본다면 영상 검색 정보와 영상 제어를 위한 모든 기능까지도 포함할 수 있다.

인터랙티브 비디오는 멀티미디어, 사용자 조작, 플레이어 등의 구성 요소와 외형적인 측면에서 HTML5 상호작용형 콘텐츠와 매우 유사하다. 그러나 인터랙티브 비디오는 프레임 단위로 움직이는 특정 이미지 위에 상호 작용 객체를 어노테이션(Annotation)하는 방식으로 비디오 파일 자체를 가공하는 것을 목적으로 한다 [11]. 연동되는 정보 역시 쇼필 정보나 URL 링크 등 매우 한정적이다. 멀티미디어와 연계된 다양한 유형의 정보와 조작 활동을 제공하고자 하는 HTML5의 상호작용 콘텐츠와는 다르다.

하이퍼비디오는 인터랙티브 비디오의 일종이다. Britta Meixner(2016)는 하이퍼비디오를 “스트리밍 비디오의 장면 정보와 비디오, 오디오, 이미지 그리고 텍스트와 같은 부가적인 정보를 연결하는 메타데이터를 포함하고 있는 것”이라 정의하였다. 하이퍼비디오는 영상의 화면 요소에 집중되었던 인터랙티브 비디오와

는 달리 메인비디오의 시점에 따라 다양한 정보를 링크하고 동기화하는 방식으로 HTML5 영상 플레이어를 통하여 서비스 할 수 있다[12]. 다만 특정 저작 도구를 통한 제작과 화면 구성 등의 제약으로 역시 다양한 형태의 콘텐츠를 표현하기에는 부족하다.

본 연구에서는 멀티미디어와 HTML5 문서의 배치를 iframe을 활용하여 구조적으로 분리하고, 시간값 데이터 연동 기능을 가진 컨테이너에 탑재하는 방식으로 설계한다. 이러한 방식으로 멀티미디어와 상호작용 HTML5 문서가 서로 독립적인 성격을 유지하면서도 최종적으로는 통합된 HTML5 웹 표준 콘텐츠로 산출될 수 있도록 한다. 또 최종 HTML5 콘텐츠는 별도의 설치 파일 없이도 웹 표준 영역에서 다양한 서비스가 가능할 뿐 아니라 특정 저작 도구나 서비스 플랫폼과의 독립성을 확보할 수 있다.

III. HTML5 iframe 기반 상호작용형 융합 콘텐츠 모형의 개발

1. 개발 모형의 구성 요소 정의

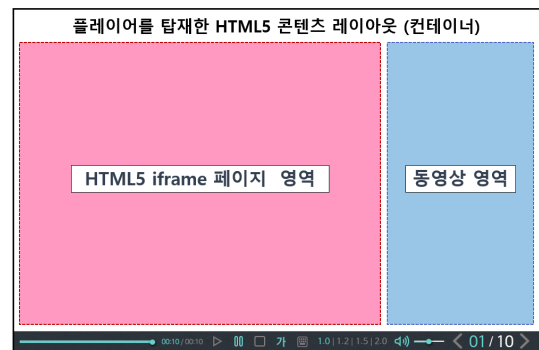


그림 2. HTML5 iframe 레이아웃과 플레이어 구성 예시

[그림 2]는 멀티미디어와 iframe의 영역을 구분 배치한 화면에 통합 플레이어가 포함된 HTML5 상호작용 콘텐츠의 화면 구성 예시이다.

콘텐츠 화면은 내용과 목적에 맞게 다양한 구성이 가능하다. 콘텐츠 제어 플레이어 역시 가로 형으로 화면 하단에 배치하는 것이 일반적이긴 하나 정해진 규격이 있는 것은 아니다[13].

자유로운 화면 구성이 가능하도록 본 연구의 콘텐츠 개발 모형에서는 콘텐츠를 구성하는 요소를 구분하였다. 모형 개발에 필요한 구성 요소는 콘텐츠의 기능과 역할, 파일 포맷 등을 고려하여 다음 3가지로 제시하였다.

○ 시간값을 가지는 멀티미디어 콘텐츠

콘텐츠의 중심 내용을 시각적, 청각적 경험을 중심으로 이끄는 요소이다. 멀티미디어는 콘텐츠 전체 시간값의 기준이 된다. HTML5의 <video>, <audio> 태그 기반의 전용 플레이어를 이용하여 사용자 상호작용 및 콘텐츠 제어 경험을 제공한다.

일반적으로 멀티미디어 콘텐츠는 전문 영상 장비나 영상 촬영이 가능한 스마트 기기를 이용하여 제작한다. 제작한 원본 영상은 별도 프로그램을 이용하여 편집하고 HTML5 표준 포맷인 MP4와 MP3 파일 포맷으로 인코딩하여 적용한다. 멀티미디어 콘텐츠의 제작과 편집은 웹 퍼블리싱과는 완전히 독립된 업무이다. 멀티미디어 콘텐츠가 HTML5 화면에 탑재되거나 링크된 이후 발생하는 수정은 원본 영상이나 음원을 별도 수정한 후 파일 자체를 교체한다.

○ iframe을 통해 제시되는 HTML5 문서

콘텐츠 내 멀티미디어의 특정 시점과 연계되는 부가 정보들은 노출되는 화면의 페이지 단위로 문서를 구분한다. 단위 페이지별로 나누어진 HTML5 문서는 분할 파일로 작성하여 정해진 시간값에 따라 iframe으로 호출된다.

분할된 HTML5 단위페이지 문서들은 문서 구조가 비교적 단순하고 작성 분량이 작기 때문에 제작과 수정, 삭제와 같은 유지보수가 용이하다. 또한 개별 페이지 내에서 HTML5와 CSS, JS를 적용하여 다양한 사용자 조작 기능을 구현 하거나 그래픽 애니메이션을 이용하여 유려한 퍼포먼스를 독립적으로 구현하는 것도 가능하다. iframe 안의 단위 페이지를 분할하여 구성함으로써 콘텐츠 기획 의도와 구현 수준을 고려하여 웹 퍼블리싱 기술 인력을 차등 배치하거나 다수의 인력들이 나누어 개발하는 등 개발 운영상의 효율을 높일 수 있다.

○ 통합 제어 플레이어가 탑재된 HTML5 컨테이너 레이아웃

콘텐츠의 기본 레이아웃이 되는 HTML5 컨테이너 페이지에는 콘텐츠 공통 기능인 HTML5 플레이어가 포함된다. 또 멀티미디어와 iframe의 배치 영역과 크기를 지정하고 병합하는 역할을 수행한다. 컨테이너 페이지의 플레이어를 통하여 멀티미디어의 시간값을 변경하고 변경된 시간값을 컨테이너 HTML5 문서에 전달할 수 있다. 플레이어는 사용자 조작으로 변경되는 특정 시점에 필요한 HTML5 단위 문서를 연동하여 iframe으로 호출할 수 있도록 동기화한다.

공통 기능인 컨테이너의 통합 제어 플레이어는 기능 요구에 따라 수정하고 변형하는 것이 가능하다. HTML5 플레이어 관련 자료가 웹상에 오픈소스로 공유되고 있으므로 이를 활용하여 개발하여도 무방하다. 다만, HTML5 콘텐츠 컨테이너 구조를 활용한 콘텐츠는 웹 표준을 준수하여 제작하고 구동되므로 일반적인 영상 편집 저작도구와 구분되며, 영상 플랫폼 서비스와는 달리 일반 웹사이트에서도 독립적으로 서비스 제공이 가능하다.

2. 멀티미디어와 HTML5 문서의 융합 설계

시간의 연속성을 가지는 멀티미디어 콘텐츠와 정지 상태의 부가 정보 콘텐츠는 제작과 편집부터 콘텐츠 출력 및 제어 방식이 모두 상이하다. 따라서 이 두 가지 요소를 하나의 콘텐츠로 융합할 수 있는 통합 제어 기능은 상호작용 HTML5 멀티미디어 융합 콘텐츠의 필수 구성 요소이다. 멀티미디어 상호작용 콘텐츠에서 사용자 직접 조작을 지원하는 플레이어는 HTML5 태그를 기본으로 구현하며, 멀티미디어의 재생, 정지, 타임라인, 자막, 음성 조절, 배속 및 구간 이동 등의 기능도 필요하면 추가하여 연동 할 수 있다.

[그림 3]은 HTML5 iframe 기반 상호작용 콘텐츠가 시간값을 연동하는 프로세스를 도식화한 것이다. 앞서 언급한 바와 같이 멀티미디어 재생 시간은 모든 콘텐츠 시간값 정보의 기준이 된다. 멀티미디어 재생 시작 시점부터 순차적으로 흘러가는 시간값 정보는 통합 플레이어에 주기적으로 입력된다. 입력된 정보는 다시 HTML5 컨테이너로 전달된다. 사전에 HTML5 컨테이

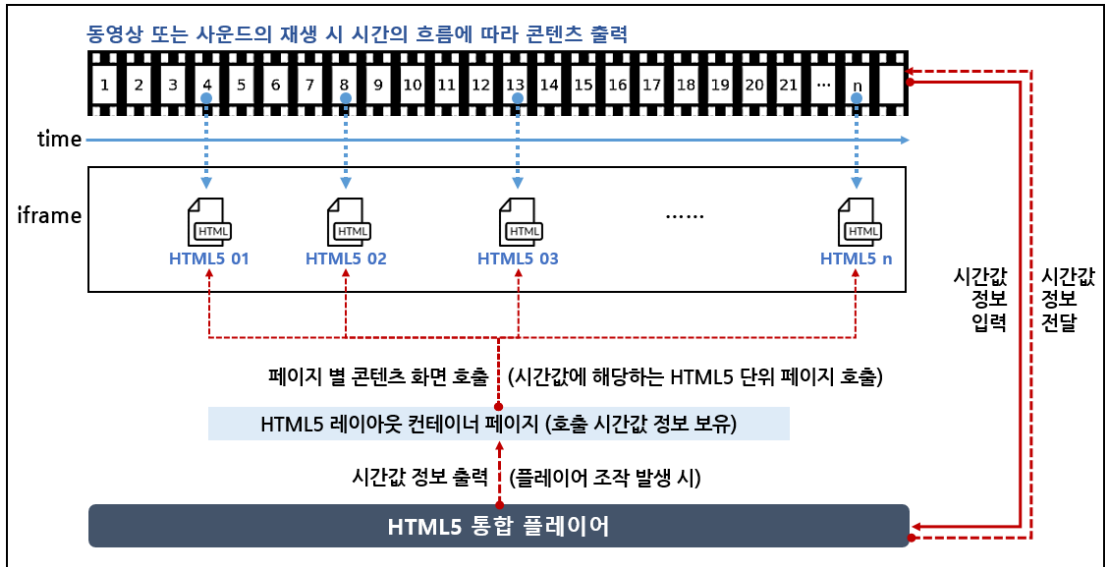


그림 3. 상호작용을 위한 HTML5 iframe기반 멀티미디어 융합 콘텐츠 구성

너에 입력된 시간값 정보와 멀티미디어에서 발생한 시간값 정보가 동일한 경우 iframe으로 그에 해당하는 HTML5 단위 페이지 문서를 호출한다.

사용자의 플레이어 조작으로 임의의 시간 값 정보가 발생될 경우 플레이어는 변경된 시간값 정보를 멀티미디어 콘텐츠와 HTML5 컨테이너로 전달한다. 멀티미디어로 전달된 시간값은 해당 시점부터 영상이 재생되도록 동기화 한다. HTML5 컨테이너로 전달된 시간값은 순차 재생 시와 동일한 방법으로 HTML5 단위페이지를 호출한다.

분리된 구조에서 동일한 시간값 정보를 공유하는 방식으로 여러 포맷의 콘텐츠들의 개발 특성과 원형을 유지하면서도 일괄 제어 하고 동기화하여 융합된 하나의 콘텐츠로 구동되도록 할 수 있다.

3. HTML5 iframe 상호작용 콘텐츠 유형

HTML5 iframe과 멀티미디어의 구분 설계 방식을 활용하여 콘텐츠를 제작하면 서비스 목적과 대상, 내용에 따라 다양한 화면 구성이 가능하다. 이에 멀티미디어와 HTML5 단위 페이지의 배치에 따라 개발 활용성이 높은 대표 유형 3가지를 선정하였다. 선정된 유형은 영상 또는 음성 강의형 이러닝은 물론 멀티미디어를 용

합하여 제작하고자 하는 모든 콘텐츠에 적용 가능하다. 편의상 유형은 멀티미디어의 종류와 화면 배치를 기준으로 명명하여 구분한다.

○ 전체 영상 유형



그림 4. 전체 영상 유형 콘텐츠 화면 예시

[그림 4]는 지정된 화면 전체에 영상 영역을 배치하고 화면 레이아웃 상위에 동영상 영역과 동일한 크기의 iframe을 배치한 형태이다. HTML5 단위 페이지는 투명 배경을 설정하여 하위에서 재생되고 있는 영상 화면이 함께 노출되도록 한다.

전체 영상 유형은 영상의 내용에 따라 화면에 제시되는 부가 정보를 다양한 그래픽 효과와 함께 제시할 수

있다. 영상 편집과 유사한 수준의 그래픽 퍼포먼스를 구현할 때 효과적이다.

전체 영상 유형은 부가 정보 내용의 수정 및 보완이 발생해도 별도의 영상 편집을 거치지 않고 HTML5 문서 코딩만으로 수정이 가능하다. 기존의 영상 편집물에서 편집 수정과 인코딩에 소요되는 작업 공수를 분산시켜 그에 따른 시간을 줄일 뿐 아니라 부가 정보의 수정과 업데이트가 매우 용이하다. 특히 영상과 부가 콘텐츠 간의 이질감이 없어 홍보마케팅 또는 시각효과가 필요한 고급형 콘텐츠 제작에 적합하다.

○ 부분 영상 유형

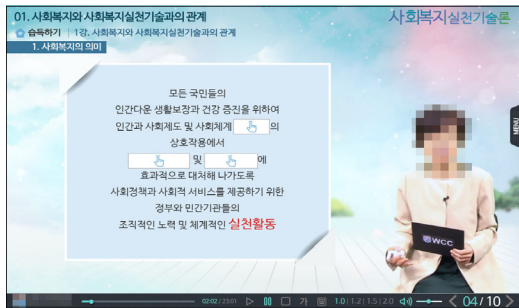


그림 5. 부분 영상 유형 콘텐츠 화면 예시

[그림 5]는 화면을 분할하여 왼쪽에 iframe, 오른쪽에 동영상을 배치하였다. 배치 방식이나 크기는 자유롭게 조절할 수 있다. 예시 화면의 영상은 크로마기로 촬영하여 iframe의 HTML5 단위 페이지의 배경과 동일 배경으로 영상을 편집하여 시각적으로 하나의 콘텐츠로 보이게 하였다. 영상의 크로마기 촬영이 어려울 경우 iframe과 영상 영역이 분리된 형태로 구성하여도 무방하다. 영상의 시간값에 따라 iframe의 부가 정보가 상시 노출되는 형식이다.

○ 사운드 유형

[그림 6]은 사운드를 활용한 콘텐츠 구성 화면 예시이다. 음성 시간값을 이용하고 HTML5 iframe 만으로 전체 화면을 구성하는 방식이다. 영상 유형과 다르게 대부분 이미지와 텍스트 중심의 정적인 화면으로 구성되며, 영상 유형과 같이 HTML5 단위 페이지의 수정과



그림 6. 사운드 유형 콘텐츠 화면 예시

업데이트가 용이하다.

HTML5 멀티미디어 융합 콘텐츠의 유형을 선정함에 있어 구성방식은 전적으로 화면 설계의 외형적인 기준을 따랐으며, 내용적인 효과성은 고려하지 않았다. 단, 이전의 플래시를 기반으로 개발되었던 콘텐츠 모형에 근거하여 멀티미디어를 활용한 상호작용 콘텐츠 유형은 대부분 적용이 가능하도록 고려하였다.

4. HTML5 iframe 상호작용 콘텐츠 모형

앞서 기술한 내용을 바탕으로 본 연구에서는 콘텐츠 구조와 실제 콘텐츠 구성 방식을 XML 표준을 활용하여 다음의 [표 4][표 5]와 같이 제시한다. 제시되는 XML 자료를 통하여 콘텐츠 개발 모형의 구조적 이해를 돕고, 향후 다양한 방식으로 콘텐츠 개발에 응용 적용할 수 있도록 한다.

[표 4]는 개발 모형의 데이터 구조를 정의한 XML DTD 문서의 예시이다. 콘텐츠를 구성하는 주요 요소인 HTML5 플레이어를 포함한 컨테이너, 멀티미디어, iframe의 연동 구조와 기본 구성 방식 등을 확인할 수 있다.

[표 5]는 정의된 XML DTD에 따라 실제 콘텐츠 개발에 적용한 XML 데이터 구현 예시이다. 본 개발 모형은 구현 단계에서는 오픈 API로 제공되는 Sax계열 파싱 프로세스를 통하여 XML과 XSLT를 JS(Javascript)로 해석하여 HTML로 출력하였다.

[그림 7]과 [그림 8]은 구분 트리형으로 구성된 XML 템스 구조를 직관적으로 파악할 수 있도록 HTML 테이블로 제시한 형태이다.

표 4. HTML5 iframe 기반 상호작용형 융합 콘텐츠 개발 모형 데이터 구조를 정의한 DTD 예시

```

<!ELEMENT options (type, setting, functions)>
  <!ELEMENT type (#PCDATA)>
  <!ATTLIST type enum (full | mix | sound) 'full'>
  <!ELEMENT setting (contents, player, video, slide)>
    <!ELEMENT contents (width, height)>
      <!ATTLIST width width_size CDATA '640'>
      <!ATTLIST height height_size CDATA '480'>
      <!ELEMENT width (#PCDATA)>
      <!ELEMENT height (#PCDATA)>
    <!ELEMENT player (x, y, width, height)>
      <!ATTLIST x pos_x CDATA '0'>
      <!ATTLIST y pos_y CDATA '0'>
      <!ATTLIST width width_size CDATA '640'>
      <!ATTLIST height height_size CDATA '480'>
      <!ELEMENT x (#PCDATA)>
      <!ELEMENT y (#PCDATA)>
      <!ELEMENT width (#PCDATA)>
      <!ELEMENT height (#PCDATA)>
    <!ELEMENT video (pos, fileinfo)>
      <!ELEMENT pos (x, y, width, height)>
        <!ATTLIST x pos_x CDATA '0'>
        <!ATTLIST y pos_y CDATA '0'>
        <!ATTLIST width width_size CDATA '640'>
        <!ATTLIST height height_size CDATA '480'>
        <!ELEMENT x (#PCDATA)>
        <!ELEMENT y (#PCDATA)>
        <!ELEMENT width (#PCDATA)>
        <!ELEMENT height (#PCDATA)>
      <!ELEMENT fileinfo (path, filename, extension)>
        <!ATTLIST path file_path CDATA './vod'>
        <!ATTLIST filename file_name CDATA "">
        <!ATTLIST extension file_ext CDATA #FIXED 'mp4'>
        <!ELEMENT path (#PCDATA)>
        <!ELEMENT filename (#PCDATA)>
        <!ELEMENT extension (#PCDATA)>
    <!ELEMENT slide (pos, fileinfo)>
      <!ELEMENT pos (x, y, width, height)>
        <!ATTLIST x pos_x CDATA '0'>
        <!ATTLIST y pos_y CDATA '0'>
        <!ATTLIST width width_size CDATA '640'>
        <!ATTLIST height height_size CDATA '480'>
        <!ELEMENT x (#PCDATA)>
        <!ELEMENT y (#PCDATA)>
        <!ELEMENT width (#PCDATA)>
        <!ELEMENT height (#PCDATA)>
      <!ELEMENT fileinfo (timeunit, path, extension)>
        <!ATTLIST path file_path CDATA './slide'>
        <!ATTLIST filename file_name CDATA "">
        <!ATTLIST extension file_ext CDATA #FIXED 'html'>
        <!ELEMENT timeunit (#PCDATA)>
        <!ELEMENT path (#PCDATA)>
        <!ELEMENT extension (#PCDATA)>
    <!ELEMENT functions (sync, volume, seek, playbackrate)>
      <!ATTLIST sync sync_show (true | false) 'true'>
      <!ATTLIST volume volume_show (true | false) 'true'>
      <!ATTLIST seek seek_show (true | false) 'true'>
      <!ATTLIST playbackrate playbackrate_show (true | false) 'true'>
      <!ELEMENT sync (#PCDATA)>
      <!ELEMENT volume (#PCDATA)>
      <!ELEMENT seek (#PCDATA)>
      <!ELEMENT playbackrate (#PCDATA)>
<!ELEMENT slides (slide*)>
  <!ELEMENT slide EMPTY>

```

표 5. HTML5 iframe 기반 상호작용형 융합 콘텐츠 개발 모형 XML 데이터 예시

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<?xml-stylesheet type="text/xsl" href="./slide_info.xsl"?>
<!DOCTYPE data SYSTEM "./slide_info.dtd">

<data>
  <!-- View구성 정보 -->
  <options name="Player_Option">
    <type>mix</type><!-- video(Full영상형) | mix(Mix형) | sound(음성형) -->
    <setting>
      <!-- 콘텐츠 전체 크기(px) -->
      <contents>
        <width>1008</width>
        <height>515</height>
      </contents>
      <!-- 플레이어 위치 및 크기(px) -->
      <player>
        <x>0</x>
        <y>480</y>
        <width>1008</width>
        <height>35</height>
      </player>
      <!-- 비디오 -->
      <video>
        <pos><!-- 위치 및 크기(px) -->
          <x>648</x>
          <y>0</y>
          <width>360</width>
          <height>480</height>
        </pos>
        <fileinfo><!-- 파일정보 -->
          <path>../user/media/mp4</path>
          <filename>03_1</filename>
          <extension>mp4</extension>
        </fileinfo>
      </video>
      <!-- 슬라이드 sync -->
      <slide>
        <pos><!-- 위치 및 크기(px) -->
          <x>0</x>
          <y>0</y>
          <width>648</width>
          <height>480</height>
        </pos>
        <fileinfo><!-- 파일정보 -->
          <timeunit>sec</timeunit> <!-- sec or ms -->
          <path>../user/sink/03_s</path>
          <extension>html</extension>
        </fileinfo>
      </slide>
    </setting>
    <functions>
      <sync>true</sync> <!-- sync show ↔ hide -->
      <volume>true</volume> <!-- volume요소 show ↔ hide -->
      <seek>true</seek> <!-- progressbar seek기능 show ↔ hide -->
      <playbackrate>true</playbackrate> <!-- 배속조절 show ↔ hide -->
    </functions>
  </options>

  <!-- 개별 sync슬라이드들 정보 -->
  <slides name="sync_Slide_List" <!-- sync슬라이드들의 공통경로 -->
    <slide time="0" file="03_tab01"/>
    <!-- 개별 sync 슬라이드들 정보 | @time: 싱크 시간, @file: 공통경로와 확장자를 제외한 파일명-->
    <slide time="4" file="03_tab02"/>
  </slides>

```

```

<slide time="492" file="03_tab03"/>
<slide time="623" file="03_tab04"/>
<slide time="805" file="03_tab05"/>
<slide time="990" file="03_tab06"/>
<slide time="1186" file="03_tab07"/>
<slide time="1348" file="03_tab08"/>
<slide time="1474" file="03_tab09"/>
<slide time="1577" file="03_tab10"/>
<slide time="1771" file="03_tab11"/>
<slide time="1879" file="03_tab12"/>
<slide time="1933" file="03_tab13"/>
<slide time="1997" file="03_tab14"/>
<slide time="2051" file="03_tab15"/>
<slide time="2111" file="03_tab16"/>
<slide time="2158" file="03_tab17"/>
<slide time="2190" file="03_tab18"/>
</slides>
</data>
    
```

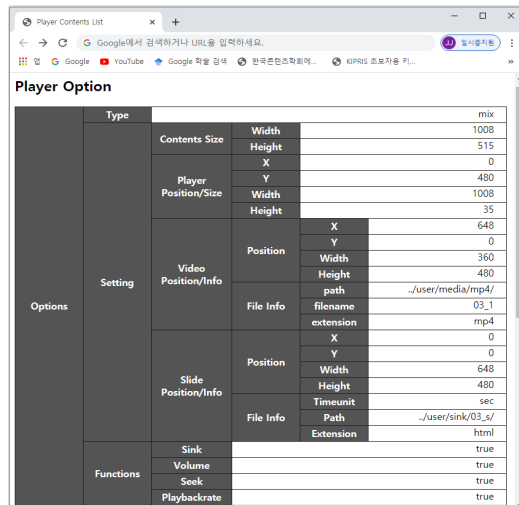


그림 7. HTML5 iframe 상호작용형 융합 콘텐츠 개발 모형의 뎁스별 구조 테이블의 웹 구현 화면 예시

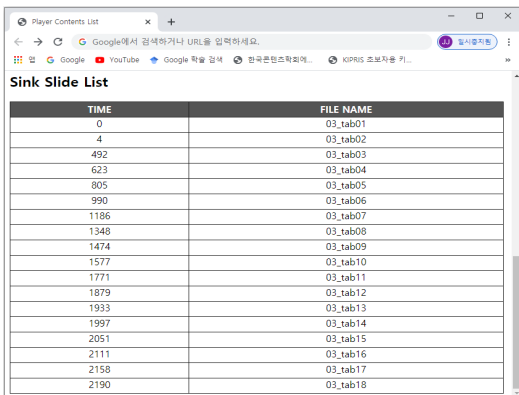


그림 8. HTML5 단위 페이지의 시간 값 정보 입력 테이블의 웹 구현 화면 예시

[그림 8]은 iframe으로 제시되는 HTML5 단위 페이지의 영상과 연계되는 시간값 정보를 HTML 테이블로 제시한 형태이다. 향후 실제 콘텐츠 구현 시 웹 개발자 이외의 개발 참여 인력이 엑셀 등의 형태로 직접 데이터를 입력할 수 있도록 활용한다.

IV. HTML5 콘텐츠 실무 개발자 대상 집단 심층 면접

1. 업무 경력에 따른 콘텐츠 개발 기간

본 연구에서 제안된 HTML5 iframe 기반 상호작용형 융합 콘텐츠 모형에 대한 실제 개발자들의 의견을 수렴해 보고자 집단 심층 면접을 진행하였다. 단순 화면 구성이 아닌 다양한 종류의 웹 프로그래밍의 구현이 요구되는 업무 특성을 고려하여 다년간 HTML5 콘텐츠를 개발해 온 전문 업체에 근무하는 실무자 7인을 대상으로 선정하였다.

면접 진행을 위하여 사전에 연구에서 제안된 HTML5 iframe 상호작용 콘텐츠(이하 HTML5 I) 개발 모형에 대하여 설명하였다. 정확한 비교를 위하여 기존 방식으로 하드 코딩하여 HTML5 단일 문서(이하 HTML5 SD)로 개발한 콘텐츠와 퍼블리싱 소스를 함께 열람하도록 하였다. 두 가지 구조의 콘텐츠 내용은 동일하다.

면접은 개인면접과 설문 후 집단 면접 하는 방식으로 진행하였다. 개인 면접 과정에서 실무 경력의 차이로

내용에 대한 이해가 부족할 경우 부연 설명을 하였으며, 설문 답변은 면접 과정에서 개발자가 직접 작성하도록 하였다. 면접은 무기명으로 진행하되, 유사 업무 근무 경력 기간이 있는 경우에는 명기하도록 하였다. 면접을 통하여 업무기간에 따라 HTML5 I와 HTML5 SD 구조에 대한 각각의 프로토타입 제작 가능 여부와 이를 활용한 시리즈물 양산 콘텐츠 개발 시 소요되는 개발 시간을 알아보고자 하였다. 또 실무 개발자가 느끼는 개발상의 이점에 대한 의견을 수렴하였다.

콘텐츠 개발과 관련하여 개발용 영상과 음원, 디자인 소스, 개발용 스토리보드 등의 리소스는 모두 제공되며, 한 차시의 콘텐츠 재생 시간은 20분, 콘텐츠 부가 정보 HTML5 문서는 20페이지로 구성함을 전제한다. 개발자 1인이 개발할 경우를 가정하고, 질문은 다음과 같다.

<질문 1> HTML5 SD / HTML5 I 방식으로 각각의 콘텐츠 프로토타입 제작이 가능한가?

(프로토타입 제작이 불가하다면 이유는 무엇인가?)

<질문 2> HTML5 SD / HTML5 I 방식으로 프로토타입 개발 시 예상 소요 기간은 각각 얼마인가?

<질문 3> HTML5 SD / HTML5 I 방식으로 개발된 프로토타입을 활용하여 한 차시 양산 콘텐츠 개발 시 개발 시 예상 소요 기간은 각각 얼마인가?

[표 6]의 [면접자 1]은 7년 이상 경력의 기술 인력으로 HTML5 웹 프로그래밍이 요구되는 교육용 콘텐츠와 게임 등의 프로토타입 개발이 가능하다. [면접자 2]는 비교적 단순한 조작 활동이 포함된 교육용 콘텐츠 프로토타입을 제작한 경험이 있다. [면접자 1]과 [면접자 2]는 HTML5, CSS, DOM, JS 및 Vue 등을 활용하여 콘텐츠와 그리기, 녹음전용 플레이어와 같은 HTML5 기반 웹 앱 형태의 제어 툴 개발 경험도 있었다. 그 외 나머지 면접자들은 1년 내외의 업무 경력으로 양산 콘텐츠 개발 작업을 주로 진행해 오고 있다. [면접자 3]은 플래시 Action Scriptor로 10여 년간 근무하다 HTML5 퍼블리셔로 전직한 개발 인력이다.

프로토타입 개발이 가능하다고 답변한 [면접자 1], [면접자 2]는 콘텐츠 구조 유형과 관계없이 개발 소요 기간을 5~7일 정도로 예상했다. [면접자 3]은 CSS의 활용에 대한 숙련도가 낮아 프로토타입 개발 시간이 더 필요하다고 답하였다. [면접자 3]은 이러한 이유로 HTML5 SD 구조에서는 프로토타입 개발에 20일 이상 소요된다고 답변했다.

숙련도가 낮은 1년 미만 업무 경력자들은 개발 구조와 관계없이 모두 프로토타입 제작 자체가 불가하다고 답변하였다. 불가한 사유는 기술 부족이며, 시간값을 다중 콘텐츠에 동시에 연동시키는 구조 자체를 구현하기가 어렵다고 답하였다. [면접자 6]은 웹 사이트 퍼블리싱 경력이 있었음에도 불구하고 플레이어와 기타 콘텐츠

표 6. HTML5 실무 개발자 집단 심층 면접 조사 결과

실무 면접자	업무 경력	HTML5 SD 구조 개발 방식		HTML5 I 구조 개발 방식	
		프로토타입 개발기간	후속 콘텐츠 양산 개발기간	프로토타입 개발기간	후속 콘텐츠 양산 개발기간
1	7년 6개월	5일	3일	5일	3일
2	2년 8개월	7일	8일	7일	4일
3	10개월 (Action Scriptor 10년 근무)	22일	12일	10일	5일
4	6개월	제작 불가	10일	제작 불가	5일
5	6개월	제작 불가	12일	제작 불가	5일
6	1년 2개월	제작 불가	20일	제작 불가	5일
7	7개월	제작 불가	10일	제작 불가	5일

츠 연동 구현과 관련한 업무 경험이 없고 소스 파악이 어려워 프로토타입 제작은 불가하다고 답변했다.

시리즈 양산 개발 기간은 콘텐츠 구조별로 소요 개발 기간의 차이가 극명하였다. HTML5 I 구조로 양산 개발할 경우 [면접자 1]을 제외하고는 1차시 분량의 양산 콘텐츠 개발에 4~5일 정도의 기간을 예상하였다. 반면 HTML5 SD 구조로 개발할 경우 1차시 양산 개발에 최소 10일에서 최대 20일까지도 소요된다고 답변하여 개인별 차이가 컸다. 길고 복잡한 HTML5 문서 작성에 있어서는 업무 경력에 따라 차이가 있겠지만 절대적으로 물리적인 시간이 소요된다는 답변이었다.

또 다른 사유는 문서의 길이가 길고 복잡해지면서 양산 개발 단계에서 혼동의 여지가 많을 뿐 아니라 수정 유지 보수에도 상당한 시간이 필요하다는 의견이 있었다.

2. 등급별 기술 인력에 따른 개발 비용

[표 7]은 [표 6]의 업무 소요 기간에 대한 일 노임단가를 적용하여 비용으로 환산한 표이다.

인건비 책정 기준은 2019년 12월 공표된 소프트웨어산업협회의 SW기술자 평균 임금을 기준하였다. 이 기준에 따라 웹 퍼블리싱 직군은 UI/UX 개발 직무 군에 속한 인건비를 적용하였다[14].

임금 기준은 상하위 구간과 평균 임금으로 나누어져

있으며, [면접자 1]은 상위 20% 평균 임금의 일평균 임금 497,210원, [면접자 2]는 전체 평균 임금의 일평균 임금 305,985원, 그 외 면접자들은 모두 경력을 고려하여 하위 20% 평균 임금의 일평균 임금 153,431원을 각각 적용하였다. 모든 임금은 기본 급, 제수당, 상여금, 퇴직급여 수당을 모두 포함한 금액이다.

프로토타입 제작에는 반드시 숙련된 기술 인력이 필요하므로 콘텐츠 구조 유형과 무관하게 유사 수준의 인건비가 투입되었다. 그러나 양산 콘텐츠 개발 비용은 HTML5 I 구조의 개발이 HTML5 SD 구조의 양산 개발에 비하여 절반 수준의 기간과 인건비가 필요한 것을 알 수 있다. 특히 HTML5 I 구조의 경우, 초보 개발자 모두 1차시 양산 콘텐츠 개발 소요 기간을 5일로 예상하여 일률적인 개발 비용의 추정이 가능하다. HTML5 SD 구조에 비해 기술 숙련도에 따른 편차가 적어 일정 계획이나 관리가 용이하고 초급 기술자 투입 운영으로 비용 절감도 기대할 수 있다.

V. 결론

1. 연구의 결과

본 연구의 HTML5 iframe 기반 상호작용 융합 콘텐츠 개발모형은 실제 콘텐츠 개발에 적용된 사례를 제시

표 7. HTML5 콘텐츠 개발 구조별 개발 소요 기간에 따른 인건비

실무 면접자	업무 경력	HTML5 SD 구조 개발 방식		HTML5 I 구조 개발 방식	
		프로토타입 개발 인건비	후속 콘텐츠 양산 개발 인건비	프로토타입 개발 인건비	후속 콘텐츠 양산 개발 인건비
1	7년 6개월	2,486,050원	1,491,630원	2,486,050원	1,491,630원
2	2년 8개월	2,141,895원	2,447,800원	2,141,895원	1,223,940원
3	10개월 (Action Scripter 10년 근무)	22일	1,841,172원	1,534,310원	767,155원
4	6개월	제작 불가	1,534,310원	제작 불가	767,155원
5	6개월	제작 불가	1,841,172원	제작 불가	767,155원
6	1년 2개월	제작 불가	3,068,620원	제작 불가	767,155원
7	7개월	제작 불가	1,534,310원	제작 불가	767,155원

하고 있다. HTML5 개발 인력의 부족과 기술 환경을 고려하여 초급 개발자도 적극적으로 개발에 참여할 수 있도록 분리 구조를 제안하였으며, 실제 개발자들을 대상으로 한 심층 면접을 통하여 개발 일정과 비용 면에서 실질적인 효과가 있는지 확인하였다.

연구 결과로 HTML5 iframe 기반 콘텐츠에 대한 프레임어와 시간값 데이터를 연동하는 기능 부분은 분리하여 XML DTD와 XML 데이터 문서로 작성하였다. 콘텐츠 개발 내용 중 대부분의 초급 개발자들이 어려워했던 시간값 데이터 연동 부분을 구조화 하여 분리함으로써 콘텐츠의 개발 난이도를 대폭 낮췄다. 또 콘텐츠 모형에 iframe을 적용하여 멀티미디어와 동기화되는 HTML5 문서를 화면 단위로 나누어 제작하도록 하였다. HTML5 문서 분할을 통하여 개발 난이도를 낮추어 초급 개발자도 무난히 개발이 가능할 뿐 아니라 여러 명이 동시에 나누어 개발하는 것도 가능하여 인력 운영의 효율성이 기대되었다.

연구 모형에 대한 실무자 대상 심층 면접에서는 HTML I 구조의 콘텐츠 양산 개발 일정에 대하여 대부분 4~5일이 소요된다고 답변하였다. HTML5 I 구조로 상호작용 콘텐츠를 개발할 경우 기존 HTML5 SD 방식과는 달리 정확한 개발 일정을 예측하여 운영 계획을 수립하는 것이 가능하였다.

같은 방법으로 산출된 개발 소요 일정에 각 등급별 인력의 일평균 인건비를 적용하여 환산한 결과 HTML5 I 방식의 콘텐츠 개발 시 HTML5 SD 방식 대비 절반 이상의 비용이 절감되는 결론을 얻을 수 있었다. 이 결과를 개발 비용의 산출이나 개발 자금 계획 수립 시의 근거 자료로 삼을 수 있을 것이다.

연구를 진행하면서 HTML5 상호작용 콘텐츠 개발에 직접적으로 참고할 만한 선행 개발물이나 공유된 기술 정보를 찾아보기가 매우 어려웠다. 또 이와 관련한 업무 프로세스나 개발 비용, 운영 등과 관련된 객관적인 정보도 부족하였다. 본 연구의 결과물의 공유를 통하여 HTML5 콘텐츠 서비스 사업자와 제작자 그리고 개발 실무자들의 어려움이 조금이나마 해소되기를 기대한다. 특히 독립적인 소규모 콘텐츠 서비스 사업자 또는 HTML5 변환 작업이 필요한 개발사 등에서 프로토타입 개발물로 적극 활용되길 바란다.

2. 연구의 한계점 및 향후 연구 방향

본 연구에서 논의된 HTML5 iframe 상호작용 콘텐츠 개발 모형은 HTML5 콘텐츠 퍼블리싱 인력과 업무를 중심으로 고안하였다. 그러나 HTML5 기반 콘텐츠 개발은 단순한 개발 기술의 변경이 아니라 전체 개발 프로세스가 변경되는 것을 의미하므로 이를 심도 있게 고려하지 못한 것은 본 연구의 한계점이다. 향후 기술 분야에 대한 연구 뿐 아니라 단계별 업무 프로세스의 정립과 효율적 운영 방식까지 통합적으로 고려한 연구가 연계될 수 있을 것이다.

본 연구에서 제안한 HTML5 iframe 상호작용 콘텐츠 개발 모형을 활용하기 위해서는 여전히 기본적인 HTML5 퍼블리싱 기술력이 요구된다. 현실적으로 숙련된 HTML5 개발 인력이 턱 없이 부족한 상황에서 실질적인 활용이 어려울 수 있다. 따라서 HTML5 상호작용 콘텐츠 개발 활성화를 위하여 퍼블리싱 전문 인력이나 기술 숙련자 없이도 콘텐츠 개발을 지원할 방안을 마련할 필요가 있다. 이에 향후 본 모형에서 도출된 데이터 구조와 구성 예시를 토대로 웹 기반 HTML5 상호작용 콘텐츠 구조 저작용 솔루션의 개발 방안에 대하여 연구하고자 한다.

참 고 문 헌

- [1] <https://www.chromium.org/flash-roadmap#TOC-Upcoming-Changes>
- [2] 홍진배, "인터넷 이용환경 개선 추진방향-웹 표준기술 (HTML5) 확산, ActiveX 대체기술 보급 중심으로," TTA저널/정보통신표준화소식, Vol.135, pp.14-19, 2011.
- [3] <https://www.koreahtml5.kr>, 한국인터넷진흥원 (KISA) 공식 HTML5 기술 지원 및 공유 홈페이지
- [4] https://medium.com/@manyong/모던_웹_스택_심층_분석-713eb1492f4e, 2016.10.10.
- [5] 이재원, "스마트교육을 위한 콘텐츠 저작 플랫폼 및 유통 포털 연구," 기초조형학연구, 제19권, 제1호, pp.411-422, 2018
- [6] 전중홍, 이승윤, "HTML5 기반의 웹 플랫폼 기술 표준화 동향," 전자통신동향분석, 제27권, 제4호, pp.83-95,

2012

- [7] 최은진, 최명숙, “이러닝 환경에서의 상호작용이 학습 효과에 미치는 영향에 관한 메타분석,” 교육공학연구, 제32권, 제1호, pp.139-164, 2016
- [8] 김영임, 유현창, “웹사이트 로딩 시간 단축을 위한 프론트엔드 프레임워크 비교분석,” 한국정보처리학회:학술대회논문집, 제25권, 제2호, pp.842-845, 2018
- [9] https://ko.wikipedia.org/wiki/디지털_콘텐츠#정의
- [10] 임명환, 허필, 박용재, “차세대 융합형 콘텐츠 분류체계 정립,” 한국콘텐츠학회:종합학술대회 논문집, pp.121-125, 2010
- [11] 윤의녕, 가명현, 조근식, “효과적인 인터랙티브 비디오 저작을 위한 얼굴영역 기반의 어노테이션 방법,” 지능정보연구, 제21권, 제1호, pp.83-98, 2015
- [12] Britta Meixner, “Download and Cache Management for HTML5 Hypervideo Players,” Conference: the 27th ACM Conference, pp.125-136, 2016
- [13] 최보아, 정종완, “e-러닝 콘텐츠의 UI Layout 분석과 학습자 요구에 따른 효율적인 UI 화면 레이아웃 연구 및 제시,” 디지털디자인학연구, 제9권, 제2호, pp.359-368, 2009
- [14] <https://www.sw.or.kr/site/sw/01/1010500000002017070309.jsp>

홍 준 석(June Seok Hong)

정회원



- 1997년 8월 : KAIST 테크노 경영 대학원 (경영공학박사)
- 1999년 ~ 2002년 : 인제대학교 경영학과 조교수
- 2003년 ~ 현재 : 경기대학교 경영 정보학과 교수

〈관심분야〉 : 빅데이터 분석과 딥러닝을 이용한 지능형 의사결정 시스템, 지능형 에이전트 시스템

김 우 주(Wooju Kim)

정회원



- 1994년 8월 : KAIST 경영과학과 (박사)
- 1996년 ~ 2002년 : 전북대학교 산업공학과 조교수
- 2002년 ~ 2004년 : 전북대학교 산업정보시스템 부교수
- 2004년 ~ 현재 : 연세대학교 산업 공학과 교수

〈관심분야〉 : 차세대 웹 기반 기술 및 응용, 웹 서비스 기반 기술 및 응용, 전자상거래 (EC) 및 E-Business, 경영정보 시스템 및 전문가시스템

저 자 소 개

이 준 정(Jun Jeong Lee)

정회원



- 2016년 2월 : 연세대학교 공학대학원 산업정보경영(석사)
- 2016년 3월 ~ 현재 : 연세대학교 융합기술경영공학 박사과정

〈관심분야〉 : HTML5 기반 상호 작용 콘텐츠, HTML5 멀티미디어 동기화 제어 플레이어, HTML5 기반 E-PUB & E-Learning 콘텐츠 및 서비스 시스템, N-Screen