



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

공학석사 학위논문

오픈시나리오 스토리 작성을 위한
투표 기반 협업 시스템



2017년 2월

부 경 대 학 교 대 학 원

IT융합응용공학과

박 상 현

공학석사 학위논문

오픈시나리오 스토리 작성을 위한 투표 기반 협업 시스템

지도교수 권 기 룡

이 논문을 공학석사 학위논문으로 제출함.

2017년 2월

부 경 대 학 교 대 학 원

IT융합응용공학과

박 상 현

박상현의 공학석사 학위논문을 인준함

2017년 2월 24일



위원장 공학박사 신 상 욱 인

위 원 공학박사 이 석 환 인

위 원 공학박사 권 기 룡 인

Voting based Collaborative System for The OpenScenario Story creation

Sang-Hyun Park

Department of IT Convergence and Application Engineering,
Graduate School,
Pukyong National University

Abstract

As the research on big data continues to improve day by day, we have already reached a level where users can take profit of various kinds of multimedia contents linked with video, sound and document work on the internet. Hence, developing a collaborative system in which users can share thoughts and contents between them, a system that can generate high quality contents based on those shared thoughts is, not only a challenging task, but also an interesting topic. By using the collaborative system services such as Amazon.com, Story Writer and Google Story Builder, we can expect real improvement of those systems concerning the fact of allowing users to create stories as fast as possible, with an assured quality and the possibility of monetizing the created contents.

Introducing a collaborative system capable of automatically generating high quality stories by just using the contents shared by the users will induce various effects such as subtitles, sound effects, background music and images for the generated story. The OpenScenario service is a collaborative system that analyzes the structure of different scenarios, which are the basis of video

production, in order to define or generate a narrative structure for the created story. The advantage of this service is that it can be easily created by ordinary users who may not be specialists of video production.

To develop a collaborative system that can automatically generate stories, a story is first created on a storyboard to make shots for constructing a scene of a video. By shot, we mean the elements necessary for story creation such as characters, conversations between those characters, sound effects and location of a person. In order to complete one scene by linking several shots, we will have a high degree of perfection if we share the roles through discussions among the collaborative users and if we add various effects necessary for the story creation.

In this paper, we use existing file sharing methods in our story creation collaborative system. The shared data between collaborative users is stored in repository. But there is a problem with this method: sometimes, conflict occurs between the saved data occasioning damage of some of the saved story files, which can be permanently lost. In the proposed method in this paper, the user moves the created story data to the story data list module before storing it in the repository in a such a way that the collaborative users can read that data in the story data list and vote on the appropriate data.

Through experiments, we confirmed that the proposed method solves the problem of collision of story data and missing data in the repository. Also, since collaborative users vote for the right story, there is no confusion about the previous shot, so the linkage between the shot is appropriate.

목 차

I. 서	론	1
II. 관련 연구		5
2.1 기존 협업 시스템 서비스		5
2.2 협업 시스템 서비스이 파일 공유 방식		11
2.3 오픈시나리오		18
2.4 프레임워크		20
III. 제안하는 스토리 작성 협업 시스템		26
3.1 스토리 작성 협업 시스템 개념		27
3.2 스토리 작성 협업 시스템 구조 설계 및 기능 분석		35
3.3 협업형 오픈시나리오 스토리 시스템		40
3.4 오픈시나리오 스토리 작성의 투표 기반 협업 기능		43
IV. 실험 결과 및 고찰		50
V. 결	론	60
참고문헌		63

그 립 목 차

그림	1. 아마존닷컴 스토리 라이터	6
그림	2. 스토리 라이터 사용방법	6
그림	3. 스토리 라이터의 진행과정에서의 (a)스토리 스크립트 작성 및 수정, (b)작성된 스토리 스크립트 초안 저장, (c) 협업 파트너에게 스크립트 데이터 공유	8
그림	4. 구글 스토리 빌더	9
그림	5. 스토리 빌더의 스토리 작성 형식	10
그림	6. Lock Modify Unlock(LMU) 방식	12
그림	7. LMU 방식의 진행 과정	13
그림	8. Copy Modify Merge(CMM) 방식	14
그림	9. CMM 방식의 진행 과정	16
그림	10. 다수의 사용자가 CMM 방식 사용 시 발생하는 문제점	17
그림	11. 오픈시나리오의 (a)시나리오와 (b)수집한 영상 편집	19
그림	12. 스프링 프레임워크의 특징	22
그림	13. MySQL Workbench 프로그램	24
그림	14. 스토리 작성 협업 시스템의 전체 구조도	26
그림	15. 전통적인 영상 제작과 스토리 작성 시스템 영상 제작	29
그림	16. OSD의 4가지 속성	30

그림 17. OSD 제어명령	33
그림 18. 오픈시나리오의 전체적인 영상 편집 구조	34
그림 19. 스토리 작성 협업 시스템 구성도	35
그림 20. 시나리오 언어 단말 편집 모듈 구성도	36
그림 21. 시나리오 언어 편집 에이전트 모듈 구성도	38
그림 22. 협업형 오픈시나리오 스토리 시스템 전체 구성도	40
그림 23. 협업형 오픈시나리오 작업 흐름도	41
그림 24. 협업형 오픈시나리오 스토리 시스템 중 투표 기반 스토리 작성협 업 시스템	44
그림 25. 오픈시나리오 스토리 작성의 투표 기반 협업 시스템	45
그림 26. 관리자모드에서의 협업 시스템 메인화면과 일반 사용자모드에서 의 협업 시스템 메인 화면	51
그림 27. MySQL을 이용한 Database 기능 구현	52
그림 28. Dashboard의 S# number의 Scenario View 화면	53
그림 29. Database의 Scenario View 데이터	53
그림 30. Scenario View의 Story Write 화면	54
그림 31. Scenario View의 Story View 화면	55
그림 32. 제안한 기법 실험의 투표 알고리즘	56
그림 33. 투표를 가장 많이 받은 스토리 데이터가 Dashboard에 등록	57

표 목 차

표 1. 스토리 라이터 서식01	7
표 2. 문서객체 모델(DOM)과 브라우저 객체 모델(BOM)	20
표 3. 전통적 영상 제작 단계와 스토리 작성 시스템 영상 제작 단계	27
표 4. OSD와 OSS의 구조와 기능	29
표 5. 기본 모델과 확장 모델 특성	30
표 6. 시놉시스 하위 속성	31
표 7. OSS의 기본 기능 및 확장 기능 특징	32
표 8. 장면 처리 명령어	33
표 9. 아마존닷컴 스토리 라이터와 구글 스토리 빌더의 문제점	58
표 10. 제안한 기법을 통해 기존의 협업 시스템 서비스의 문제점 개선 ...	59

I. 서론

우리가 현재 살아가고 있는 인터넷 시대의 대중화와 정보기술의 발달은 언제 어디서나 네트워크에 접속할 수 있도록 모바일 기술이 발전하였다. 또한 점점 방대해지는 데이터를 쉽게 활용하기 위해 빅데이터(Big Data)에 대한 많은 연구가 진행되고 있다[1]. 빅데이터를 이용한 데이터 공유 시스템은 데이터 정제, 활용성 분석, 관리 등의 모듈별 시스템 중점으로 데이터들이 공유되고 있다[2,3]. 특히 웹 상에서 영상이나 음성, 문서 작업 등이 연계되어 다양한 멀티미디어 콘텐츠(Multimedia Contents)를 활용하는 방식으로 관심이 높아지고 있다[4]. 데이터의 활용이 방대해지고 이에 따른 콘텐츠의 다양성과 활용성이 증가함에 따라 콘텐츠를 사용하는 사용자가 혼자서는 콘텐츠를 작성하기 보다는 다른 사용자와 협업을 통한 협업 시스템(Collaboration System)을 이용함으로써 고품질의 콘텐츠를 생성할 수 있다[5]. 콘텐츠 사용자 혼자서 다양해진 콘텐츠를 작성하는 것은 다수의 협업 사용자가 참여한 콘텐츠보다 아이디어가 부족하다. 그런 문제점을 보완하기 위해 다른 협업 사용자들을 콘텐츠에 참여시켜 콘텐츠 사용자들 간의 정보 공유와 역할 분담을 나누어 체계적으로 콘텐츠를 생성함으로써 생성시간의 단축과 완성도 면에서도 좋은 결과를 기대할 수 있다[6,7]. 스토리 작성 협업 시스템 서비스는 다수의 콘텐츠 사용자들이 협업 과정을 통해 스토리 작성을 하는데 있어 고품질적이고 다양한 콘텐츠를 생성한다[8].

대표적인 협업 시스템 서비스들로는 아마존닷컴의 스토리 라이터(Story Writer)[9], 구글의 스토리 빌더(Story Builder)이 있다[10]. 아마존닷컴의 스토리 라이터는 아마존닷컴에서 제공하는 사용자간의 협업 시스템 서비스

로 사용자의 계정에 문서를 작성할 수 있도록 스크립트를 제공하고 있다. 작성된 스크립트는 자신의 계정에 저장하거나 다른 사용자와 작성된 스크립트를 공유하여 협업할 수 있는 시스템이다[11]. 또 다른 예인 구글의 스토리 빌더는 구글에서 제공하고 있는 사용자간 협업 시스템 서비스이다. 여러 사용자들이 하나의 스크립트를 공유하여 사용자들 간의 순서를 정하고, 정해진 순서대로 공유된 스크립트를 수정하는 협업 시스템이다.

오픈시나리오(OpenScenario)는 영화, 드라마 등의 영상물 제작의 기초가 되는 시나리오(scenario)에 대한 구조를 분석하여 스토리 표현 구조를 정의하는 협업 시스템 서비스이다[12]. 이 협업 시스템 서비스를 사용함으로써 영상물 시나리오를 생성하는데 있어 간략화된 구조와 쉬운 인터페이스를 가지고 있어 시나리오 작가와 같은 전문가가 아닌 일반 사용자들도 영상물 시나리오를 작성할 수 있는 구조를 지니고 있다.

스토리 작성 협업 시스템 서비스의 특징은 콘텐츠를 제작하기 위하여 스토리보드를 작성하여 영화, 드라마와 같은 영상물의 장면 신(Scene)에 대한 여러 샷(Shot)들이 생성한다. 샷에는 등장인물, 대화의 내용들, 음향, 인물의 위치, 표정 등을 스토리 작성을 통해 생성된다[13]. 정해진 순서의 샷들은 콘텐츠 사용자들이 협업 과정에서 역할을 분담하고, 담당 스토리를 작성하여 생성된 샷들을 연계시켜 하나의 장면을 완성하게 된다. 이러한 스토리 작성 협업 시스템이 상용된다면 다양한 영상물 콘텐츠를 제작하는데 있어 높은 완성도를 가지게 될 것이다.

기존의 협업 시스템들은 파일 공유 방식인 Lock Modify Unlock(LMU) 방식과 Copy Modify Merge(CMM) 방식을 사용하고 있다[14]. LMU 방식은 저장소의 사용 권한을 특정 사용자에게 부여하여 저장소의 읽기/쓰기를 할 수 있도록 한다. 권한을 받지 못한 다른 사용자들은 저장소 접근에 제한 받게 되며, 우선 순위 권한 사용자가 데이터를 업로드하는 동시에 권한

이 풀려 다른 사용자들도 저장소의 데이터를 읽을 수 있게 된다. 이러한 LMU 방식의 장점은 사용 권한에 의해 강력한 보안성을 지니고 있다[15]. 문제점으로는 저장소 사용 권한에 따른 다수의 사용자가 사용하기에는 부적합하다. 그리고 우선 권한 사용자가 데이터를 읽기만 하고 업로드를 하지 않을 경우에 다른 협업 사용자들은 무한정 대기 상태인 교착상태에 빠지게 된다. CMM 방식은 LMU 방식의 문제점들을 보완하기 위해 나왔으며, 저장소에 데이터를 업로드 시 모든 협업 사용자들에게 저장소에 업로드된 데이터를 복사하여 전송하는 방식이다. 장점으로는 LMU 방식의 문제점이었던 다수의 사용자가 사용하기에는 부적합했던 것을 다수의 사용자가 사용 가능하게 되었다. 하지만 CMM 방식에도 문제점이 있는데, 첫 번째는 저장소는 데이터들이 업로드 될 때마다 기존의 데이터에 덮어쓰기 때문에 원본 데이터의 보존이 불가능하다. 두 번째는 다수의 협업 사용자들이 CMM 방식을 사용할 경우, 저장소에 데이터들을 동시에 업로드 했을 때, 업로드 되는 데이터들 간의 충돌로 인해 저장소에 업로드 되어야 할 데이터가 누락되거나 충돌로 인해 훼손된 데이터가 저장소에 업로드 되는 경우가 발생한다.

하지만 기존의 협업 시스템의 파일 공유 방식을 스토리 작성 협업 시스템 서비스로 구현할 경우 위에서 언급한 LMU 방식의 교착 상태 발생과 CMM 방식의 원본 데이터 보존 불가, 데이터 간의 충돌로 인한 데이터 누락 및 훼손 문제점들이 발생한다. 이러한 문제점들을 해결하기 위하여 투표 스토리 작성 협업 시스템 기법을 제안하고자 한다.

본 논문에서는 투표 방식의 오픈시나리오 기반 스토리 작성 협업 시스템을 제안한다. 제안한 시스템은 CMM 방식의 데이터를 복제하여 모든 협업 사용자들에게 전송하는 과정이지만 협업 사용자들이 데이터를 수정하여 업로드하는 경우에 메인 저장소에 저장되지 않고 투표 인터페이스에 저장되

어 업로드 데이터들을 수집한다. 협업 사용자와 메인 작성자는 투표 인터페이스에 수집된 업로드 데이터들을 확인한 후 해당 저장소에 적절한 데이터를 선정하여 메인 저장소에 업로드한다.

제안한 투표방식의 협업 시스템은 CMM 방식의 원본 데이터 보존 불가, 데이터 누락 및 훼손 문제점을 보완하기 위한 협업 시스템이다. 제안 기법을 사용했을 때 발생하는 장점으로는 첫 번째, 업로드 되는 데이터의 정보를 일일이 확인할 필요가 없다. 두 번째, 업로드되는 데이터들간의 충돌을 방지하여 데이터가 누락되거나 훼손되는 것을 차단할 수 있다. 세 번째, 해당 저장소에 투표를 통하여 데이터가 업로드 되므로 협업 사용자들 간의 공정성이 부여된다.

제안 기법을 실험한 결과로는 협업 사용자들 간의 스토리 데이터를 확인 이후, 투표를 통하여 해당 스토리 구간 저장소에 적절한 스토리 데이터를 선정하여 업로드 하였다. 투표를 통한 선정 진행 방식으로 업로드 되는 스토리 데이터들 간의 충돌을 방지하여 데이터가 누락되거나 훼손되는 경우를 차단할 수 있었다. 제안한 시스템은 기존 협업 시스템 서비스에 도입하여 스토리 작성 협업하는데 있어 고품질, 다양성, 생성 시간 단축 등 여러 가지 이점을 가질 수 있다.

본 논문의 구성으로 2장에서는 본 논문에 대하여 이해를 향상시키기 위한 관련 연구들에 대하여 살펴보고, 3장에서는 본 논문에서 제안하는 투표 기반 스토리 작성 협업 시스템 방식에 대해 상세하게 서술 하였다. 4장에서는 제안한 스토리 작성 협업 시스템 방식의 사용하여 개선된 협업 시스템을 구현한다. 마지막으로 5장에서는 제안 논문의 결론을 맺으며 향후 연구 방향 및 개선점에 대하여 제시한다.

II. 관련 연구

본 절에서는 대표적인 협업 시스템 서비스들인 아마존닷컴의 스토리 라이터와 구글의 스토리빌더를 살펴본다. 그리고 기존의 협업 시스템 서비스들의 파일 공유 방식에 대해 알아보고, 스토리 작성 콘텐츠에 적용 시 발생하는 문제점들에 대하여 분석한다. 마지막으로 다른 협업 시스템 형식의 서비스인 오픈시나리오에 대해 살펴본다.

2.1 기존 협업 시스템 서비스

2.1.1 아마존닷컴 스토리 라이터

아마존닷컴은 전자책 콘텐츠 위주의 인터넷 온라인 서점으로 출발하여 대규모 데이터베이스 운영 기술을 활용하여 클라우드 컴퓨팅 서비스 분야로 사업을 확장하여 최근에는 데이터베이스 기술, 이용자 콘텐츠의 활용방법, 방대한 데이터를 저장하는 방법, 다양한 디지털 콘텐츠 확보 및 이용방법을 결합하여 새로운 서비스들을 제공하고 있다[16]. 이러한 새로운 서비스들 중에 스토리 라이터라는 클라우드 기반의 무료 시나리오 소프트웨어 프로그램을 제공하고 있다.

스토리 라이터는 그림 1에서와 같이 아마존닷컴의 사용자들 간의 시나리오, 소설과 같은 문서 파일을 작성할 수 있도록 스크립트를 제공하고 본인 계정 또는 협업하고 있는 다른 사용자의 계정에 저장하여 제공되는 스크립트를 협업을 통하여 작성할 수 있는 시스템이다.



그림 1. 아마존닷컴 스토리 라이터

아래의 그림 2는 스토리 라이터의 스크립트를 사용하는 방법에 대해 설명하고 있다. 문서 작성 방식을 사용하여 스크립트 작업을 작성하는 방법으로 시나리오의 대본을 작성할 때 서식을 편의에 따라 지정할 수 있어 효율적이다.

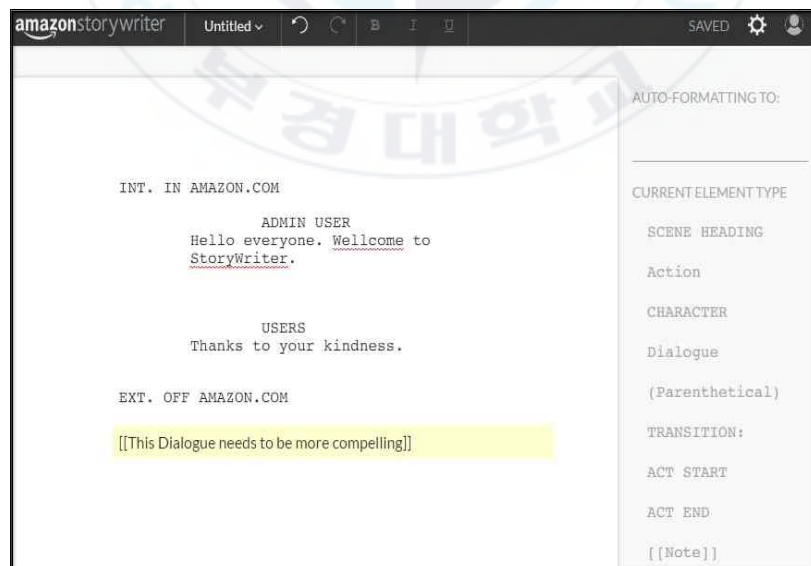


그림 2. GIS 벡터맵 데이터 구조

표 1. 스토리 라이터 서식

서 식	기 능
Scene heading	· 시나리오 장면에서 사용되는 내부장면(INT), 외부 장면(EXT)을 작성할 때 사용
Character	· 시나리오 등장인물의 이름을 표기할 때 사용
Dialogue	· 등장인물의 대화를 나타내는 서식
Parenthetical	· 등장인물의 표정 또는 행동을 작성할 때는 괄호 (·)를 사용
Transition	· 장면이 끝나면서 점점 어두워지는 FADE OUT을 나타내는 서식
Note	· 시나리오의 주의사항 또는 참고를 작성할 때 사용

표 1은 스토리 라이터에서 스토리를 작성할 때 자동으로 서식이 지정된다. 다음 행으로 넘어갈 때 각 서식의 종류를 바꾸어 선택하거나 그대로 유지하여 할 수 있다. 작성하고 있는 스토리 라이터 데이터의 진행 상황을 표시하여 저장되는 라이브러리에 혼동을 주는 것을 막을 수 있다. 그리고 협업 스크립트를 작성하고 있는 경우, 상대방의 계정으로 현재까지 작성된 스토리 라이터 데이터를 보내어 협업 상대가 읽기/쓰기를 할 수 있다. 스토리 라이터의 장점으로서는 컴퓨터 이외에도 다른 기기에서도 작업 스크립트를 액세스하여 읽기/쓰기 과정을 진행 할 수 있다.

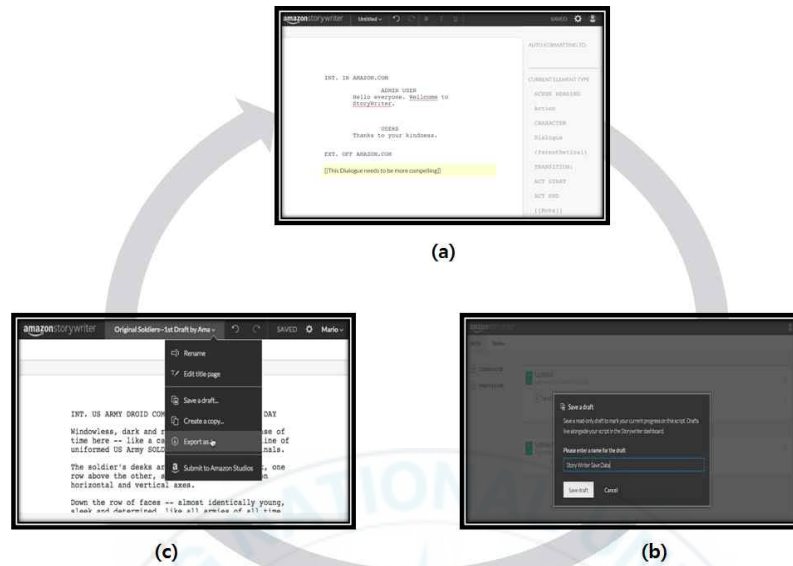


그림 3. 스토리 라이터의 진행과정에서의
(a)스토리 스크립트 작성 및 수정, (b) 작성된 스토리 스크립트 초안 저장
및 (c) 협업 파트너에게 스크립트 데이터 공유

아마존닷컴의 스토리 라이터는 위의 그림 3과 같이 스토리 스크립트를 작성 및 수정 이후에 스크립트 초안을 저장하고, 협업 파트너에게 작업 스크립트를 전송하는 진행과정을 통하여 스토리 작성 협업 콘텐츠 서비스를 제공하고 있다.

2.1.2 구글 스토리 빌더

구글은 ‘페이지 랭크’라는 독자적인 검색 알고리즘을 개발하여 인터넷 검색 시장을 장악하고 있는 세계 최대 인터넷 검색 서비스 회사이다. 구글은 인터넷 검색의 다양한 서비스를 제공하며, 그 중에 스토리빌더가 포함된다.

스토리빌더는 그림 4에서와 같이 사용자들이 한 스크립트에 대화형식으로 작성하는 협업시스템 서비스이다. 본인이 협업 작업에 관하여 작성을 하면, 채팅하고 있는 다른 협업 사용자가 응답해주는 형식이다. 1:1 채팅 형식의 협업뿐만 아니라 협업 그룹 채팅 방식도 가능하며, 다자간의 문서 공유와 수정이 가능하다. 다른 협업 시스템 서비스보다 스토리빌더는 포맷이 쉽고 단순하여 스토리를 작성하는데 시간 소요가 짧고 조작이 간편하다.



그림 4. 구글 스토리 빌더

스토리 빌더를 작성하기 위해서는 그림 5와 같이 등장인물들을 생성하여 시나리오 대사와 같은 스토리 작성 형식으로 작성한다. 생성한 등장인물의 대사를 작성하면 다른 등장인물이 대사를 이어 작성한다. 스토리 작성은 채팅형식으로, 동영상을 통하여 생성된 등장인물의 대사를 작성한 과정을 보여준다. 협업을 통한 스토리 작업일 경우, 다른 사용자가 스토리를 수정할 때, 동영상을 통해 수정 내용을 확인할 수 있다.

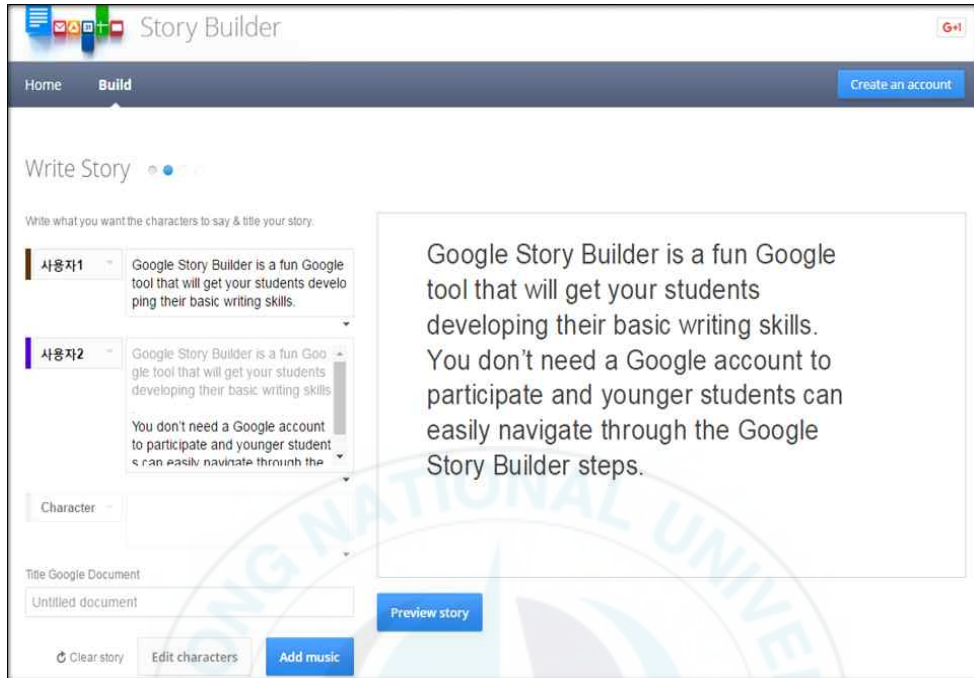


그림 5. 스토리 빌더의 스토리 작성 형식

스토리 빌더는 작성된 스토리 스크립트에 음향 효과를 삽입하여 효과적인 스토리 콘텐츠를 작성할 수 있다. 구글에서 제공하고 있는 몇 가지 음향만 제공하고 있지만 차후에 다양한 음향효과나 작성한 스토리의 녹음 파일을 업로드 하여 스토리를 작성하는데 품질적인 향상과 높은 완성도를 기대할 수 있다. 완성된 스토리는 동영상 링크를 걸어 제작과정을 동영상으로 편집하여 보여준다. 다른 협업 사용자들에게 링크 주소를 알려주거나 스토리 빌더파일을 공유하여 스토리를 협업 과정을 통하여 작성한다.

2.2 기존 협업 시스템의 파일 공유 방식

기존의 협업 시스템 서비스들은 파일 공유 방식으로 Lock Modify Unlock 방식과 Copy Modify Merge 방식으로 사용되고 있다. 스토리 작성 협업 시스템을 생성할 시 위의 두 가지 파일 공유방식을 사용하게 될 경우, 협업 사용자들의 교착 상태, 원본 데이터 보존 불가, 데이터 충돌로 인한 누락 및 훼손 등의 문제점들이 발생하게 된다[17].

2.2.1 Lock Modify Unlock(LMU) 파일 공유 방식

Lock Modify Unlock 파일 공유 방식은 그림 6에서와 같이 메인 작성자가 생성한 스토리 작성 콘텐츠의 메인 저장소에 협업 사용자들이 접근하기 위해서는 접근 권한을 부여받아야 한다. 저장소에 최초로 데이터를 읽은 협업 사용자에게 우선 순위 권한을 부여하고 비권한 협업 사용자들은 저장소의 데이터를 읽지 못한다. 권한을 가진 협업 사용자가 읽은 데이터를 수정하여 업로드하면, 그때서야 접근 권한이 풀려 다른 사용자들도 저장소에 접근 가능할 수 있게 된다.

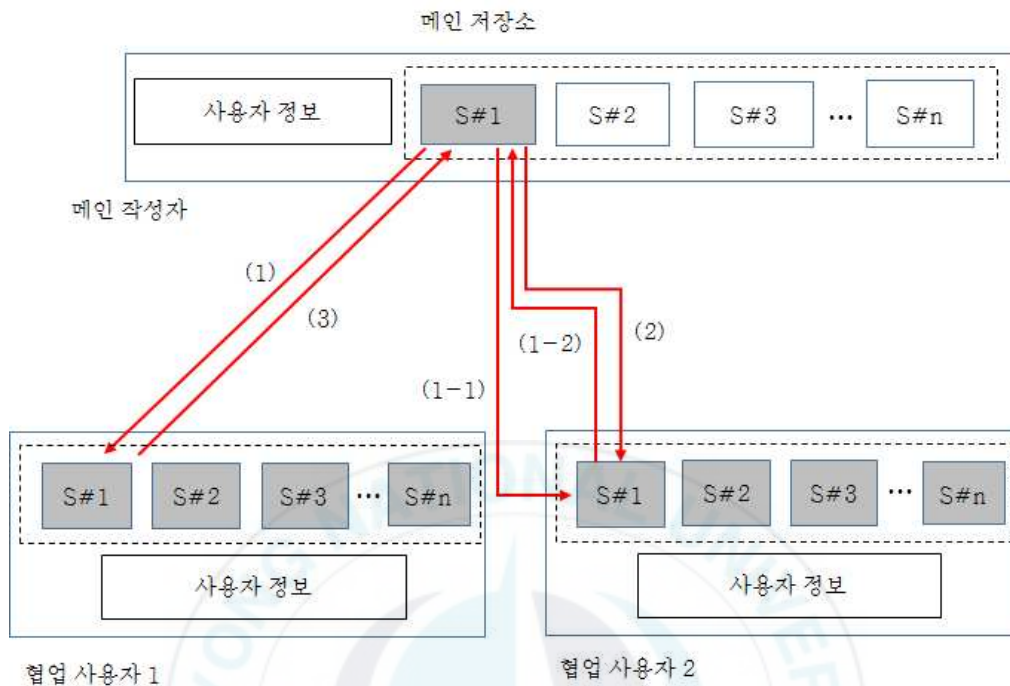


그림 6. Lock Modify Unlock(LMU) 방식

그림 7에서는 LMU 방식의 진행 과정을 구분하였다. 첫 번째 그림에서는 메인 저장소에 최초로 접근하여 데이터를 읽은 협업 사용자1이 저장소의 우선 권한을 부여 받고 다른 사용자들은 접근하지 못하도록 잠긴 상태가 된다. 두 번째 그림에서는 권한이 없는 다른 협업 사용자들이 잠긴 상태인 메인 저장소에는 데이터를 읽지 못한다. 세 번째 그림에서는 권한을 가진 사용자가 읽은 데이터를 수정하여 저장소에 업로드한다. 이 때 사용 권한 상태를 해지하면서 Lock 상태가 풀리게 된다. 마지막 그림에서는 잠긴 상태가 풀려 다른 사용자들도 접근이 허용되어 저장소의 데이터를 읽어 올 수 있게 된다. Lock Modify Unlock 방식의 장점은 권한을 부여함으로써 저장소의 강력한 안정성이 보장된다. LMU 방식의 문제점은 첫 번째는 우선 순위 권한 방식에 의해 협업 사용자가 두 명 이상의 데이터를 수정하기

어렵다. 두 번째는 권한을 가진 사용자가 Read만 수행하고 업로드를 하지 않은 상태로 유지한다면 다른 비권한 사용자들은 무한정 대기상태인 교착 상태에 빠지고 만다. 세 번째는 위의 문제인 교착 상태를 염려하여 사용자들은 각각의 다른 장면을 분담하여 업로드를 하면 앞의 장면 내용 연계가 일치하지 않는다. 마지막으로, LMU 방식을 수행 완료하는데 많은 소요시간이 걸린다.

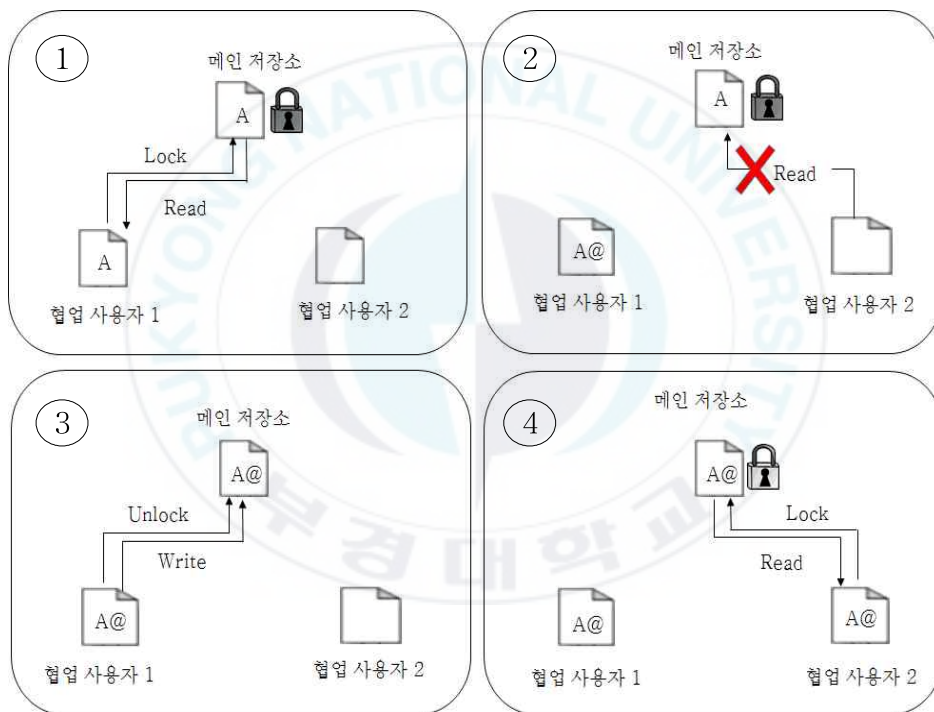


그림 7. LMU 방식의 진행 과정

2.2.2 Copy Modify Merge(CMM) 파일 공유 방식

Copy Modify Merge 방식은 그림 8에서와 같이 LMU 방식의 문제점인 두 명 이상의 사용자가 사용하는데 어려웠던 것을 다수의 사용자가 사용 가능하도록 제안된 방식이다. 저장소에 데이터가 입력되면 협업에 참가한 모든 사용자들에게 저장소에 업로드된 데이터를 복사하여 전송한다. LMU의 우선순위 권한이 없기 때문에 모든 협업 사용자들이 저장소에 업로드할 수 있다.

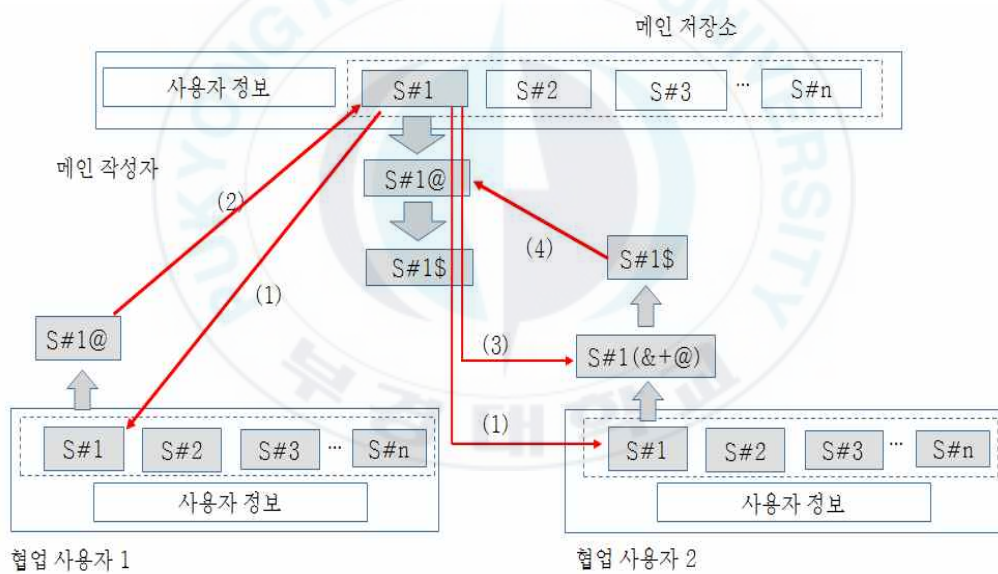


그림 8. Copy Modify Merge 방식

그림 9에서는 CMM 방식의 진행 과정을 나타낸다. 메인 저장소에 데이터가 저장될 때, 데이터가 복사되어 모든 협업 사용자에게 전송된다. 전송받은 데이터를 자신만의 데이터로 수정한다. 협업 사용자는 수정한 데이터를 메인 저장소에 자유롭게 업로드할 수 있다. 메인 저장소에 데이터

가 업로드되면 업로드된 데이터를 다시 복사하여 협업 사용자들에게 전송하게 된다. 수정 중인 데이터와 업로드 데이터를 합쳐 새로운 데이터를 생성한다. 새롭게 생성된 데이터를 메인 저장소에 업로드하면 저장소는 업로드 데이터를 다시 복사하여 다른 협업 사용자들에게 복사하여 전송하는 방식이다.

Copy Modify Merge 방식의 장점으로는 두 명 이상의 사용자들이 데이터를 수정 가능하며, 우선 순위 권한이 존재하지 않기 때문에 협업 사용자들이 교착 상태에 빠지는 일은 발생하지 않는다. 하지만 CMM 방식에도 문제점이 발생한다. 첫 번째는 저장소에 기존의 원본 데이터가 업로드 되는 데이터에 덮어쓰기 때문에 원본 데이터 보존이 불가능 하다. 데이터들이 업로드되는 과정에서 협업 사용자들이 복사된 원본 데이터를 읽고 싶어도 업로드된 데이터로 덮어쓰기 때문에 원본 데이터를 읽을 수 없는 경우가 발생한다. 두 번째는 LMU 방식과 마찬가지로 수행 완료하는데 많은 소요 시간이 걸린다.

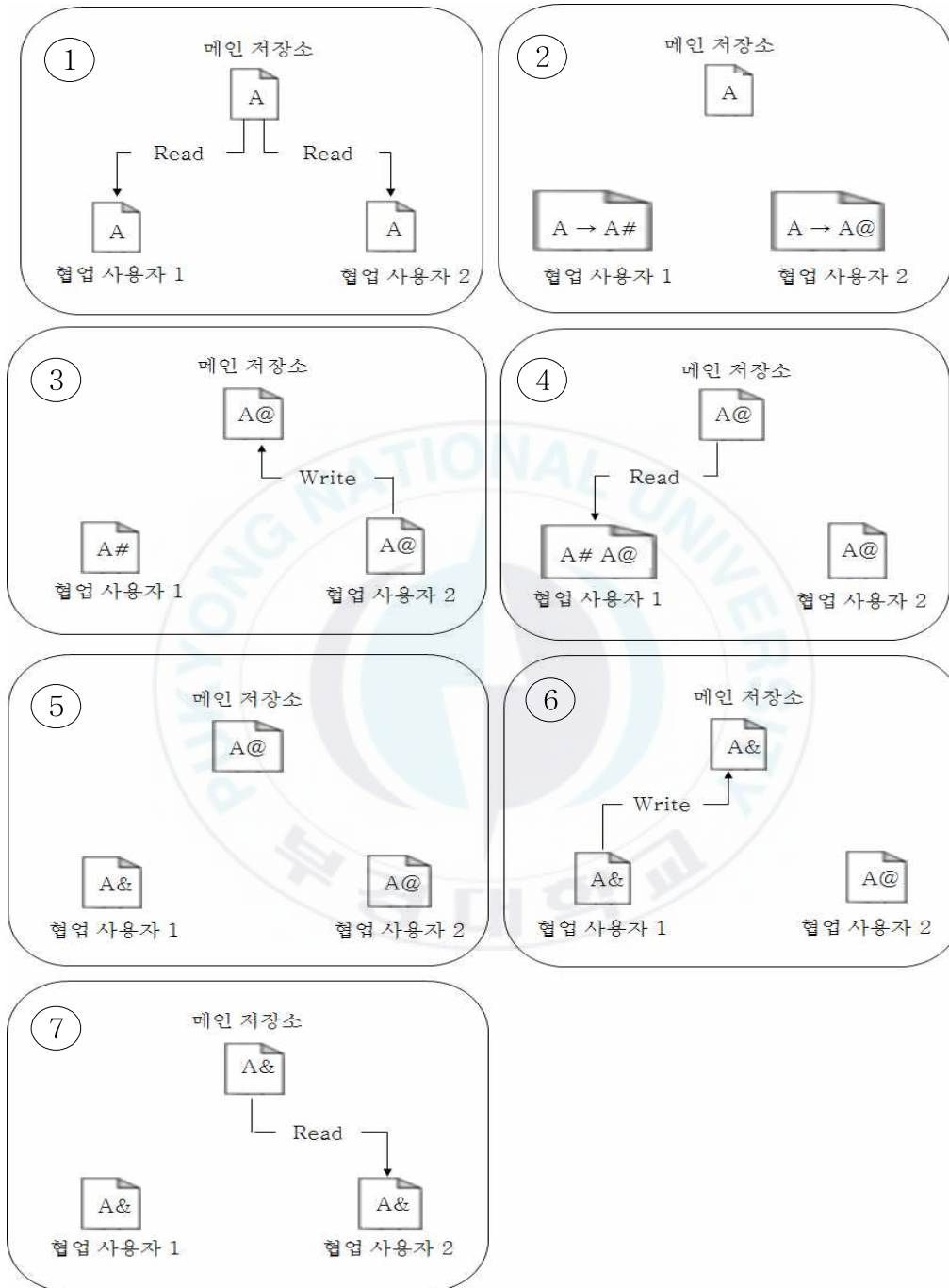


그림 9. CMM 방식의 진행 과정

그림 10에서는 다수의 협업 사용자들이 CMM 방식을 사용하게 될 경우, 한 저장소에 여러 데이터들이 동시에 업로드되어 데이터들 간의 충돌로 인하여 데이터가 누락되어 저장소에 업로드되지 못하거나 저장소에 훼손된 데이터가 업로드되는 경우가 발생한다.

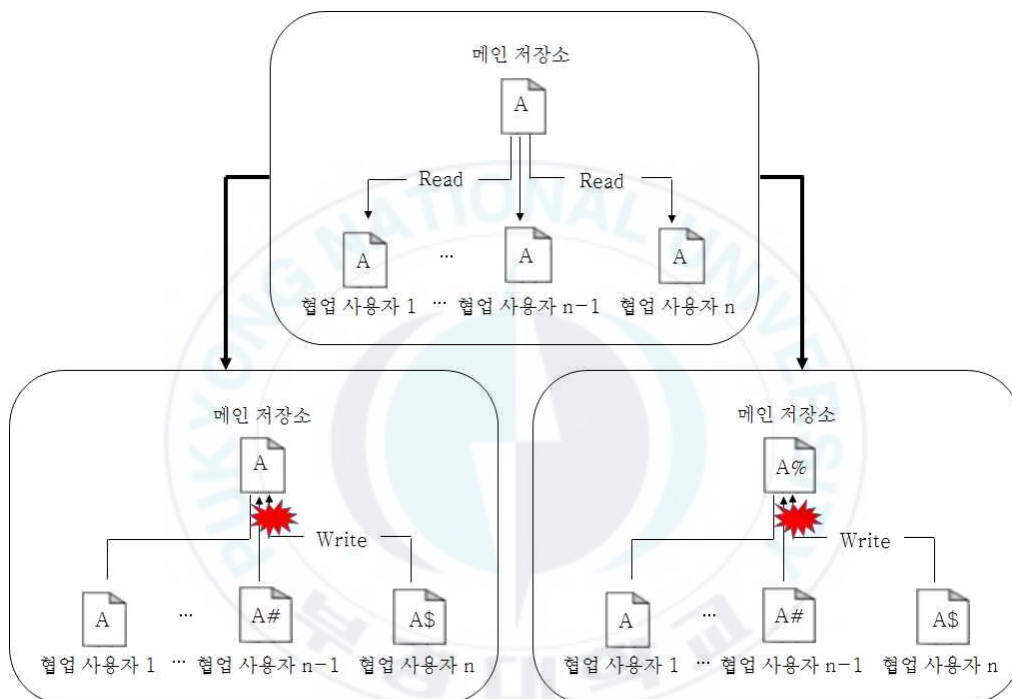


그림 10. 다수의 사용자가 CMM 방식 사용 시 발생하는 문제점

2.3 오픈시나리오

오픈시나리오는 사용자가 생성한 스토리에 음향, 영상 등의 부합된 콘텐츠를 웹상에서 검색하여 이를 활용한 콘텐츠를 재생성 및 품질 향상 시킨다. 일반적인 영상 제작일 경우, 작성된 시나리오를 바탕으로 영상 촬영과 편집을 걸쳐 최종 영상이 제작된다[18]. 하지만 오픈시나리오 서비스는 영상 촬영을 하지 않고 웹상에 존재하는 많은 영상 데이터를 활용한 영상 수집으로 대체한다. 그림 11에서는 오픈시나리오 서비스는 작성된 스토리에 따라 적절한 영상을 웹상에서 검색, 수집하고, 수집된 영상에 등장인물의 행동, 음향, 컴퓨터그래픽 등의 다양한 효과를 추가하여 영상을 편집 이후에 최종 영상을 출력한다[19,20].

오픈시나리오는 시나리오, 영상 및 편집 요소들을 포함할 수 있도록 구조화 한다. 시나리오 작가의 관점에서 스토리를 기술하는 요소를 포함한 OSD(OpenScenario Descriptors)와 감독의 관점에서 촬영요소 및 편집 요소들을 포함한 OSS(OpenScenario Scripts)이다[21]. OSD는 콘텐츠 제작 시 시나리오 쓰기 및 촬영의 관점에서 등장인물이나 배경 등을 기술의 집합이다. OSS는 콘텐츠 제작 시 편집 부분에서 자막, 이미지, 음향효과, 배경 음악 등을 추가 또는 변형에 대한 스크립트의 집합이다.

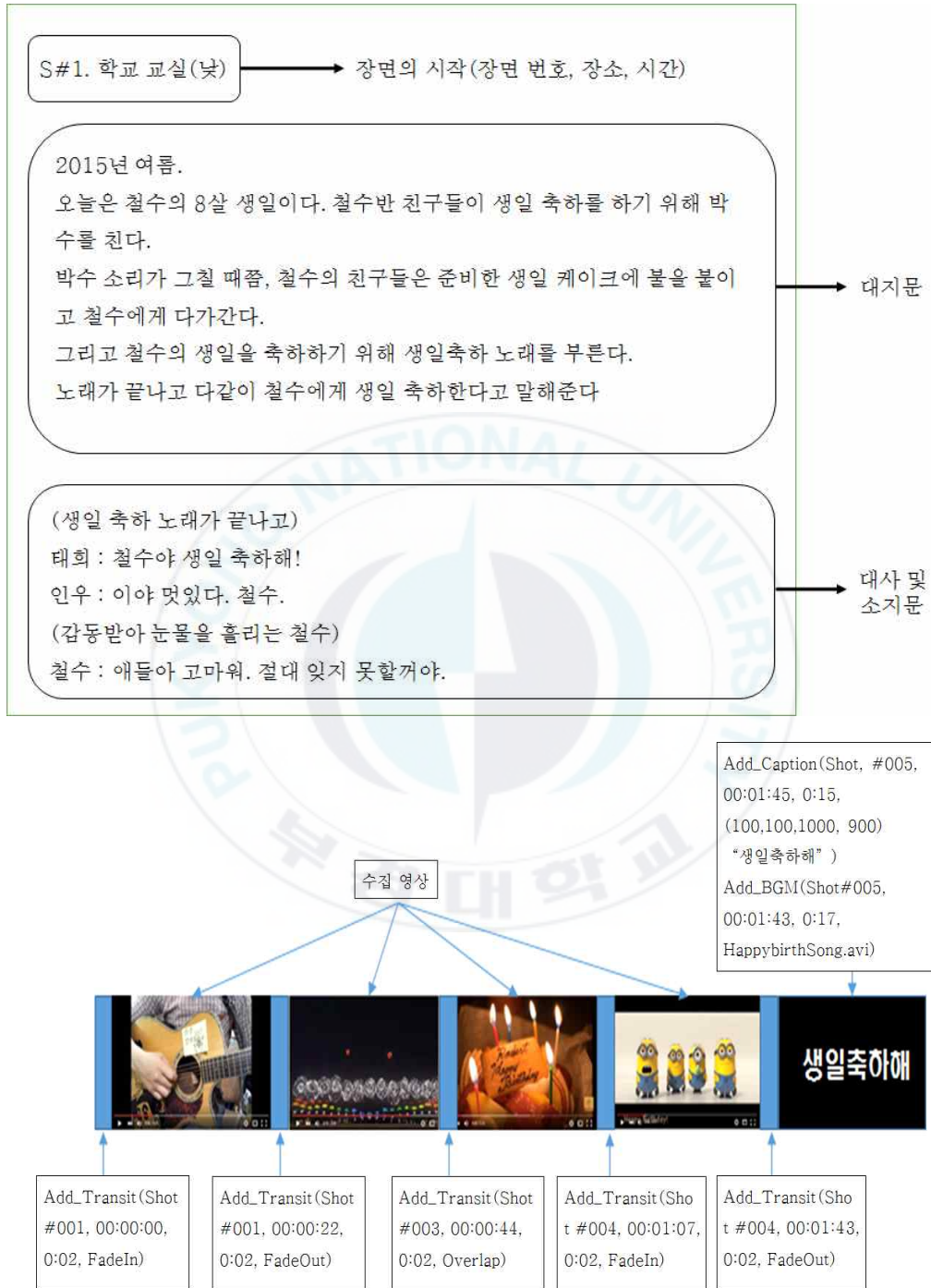


그림 11. 오픈시나리오의 (a)시나리오와 (b)수집한 영상 편집

2.4 프레임워크

2.4.1 Java Script

자바 스크립트(Java Script)는 웹 페이지와 상호작용 하도록 만들어진 스크립트 언어다. ECMA-262에서 정의하는 ECMAScript의 핵심 기능을 제공하고 있으며, 표 2에서는 문서 객체 모델을 통해 웹 페이지 콘텐츠를 조작하는 메소드와 인터페이스를 제공하며, 브라우저 객체 모델을 통해 브라우저와 상호작용하는 메소드와 인터페이스를 제공한다. ECMAScript는 자바 스크립트의 문법, 데이터형, 명령어, 키워드, 예약어, 연산자, 객체 등에 대하여 정의한다[22].

표 2. 문서객체 모델(DOM)과 브라우저 객체 모델(BOM)

문서객체 모델(DOM)	브라우저 객체 모델(BOM)
• XML을 HTML에서 사용할 수 있도록 확장한 애플리케이션 프로그래밍 인터페이스 • 전체 페이지를 노드의 계층 구조로 변환하고 다양한 타입의 노트로 표현 가능 • 문서를 표현하는 트리를 생성하고 개발자는 이를 통해 문서의 콘텐츠와 구조를 자유롭게 수정 가능	• 브라우저 창에 접근하고 조작할 수 있는 인터페이스 • 표시된 페이지와는 별개의 컨텍스트에서 브라우저와 상호작용 가능 • 브라우저의 창을 새로 띄우거나 움직이거나 크기 조절 및 닫음

자바 스크립트는 스크립트 언어이며, 웹에서 사용하는 인터프리터 언어다. 컴파일이 필요없으며, HTML 웹 페이지를 통해 스크립트를 삽입하면 웹 브라우저를 통해서 동작하게 된다. 자바스크립트의 특징은 HTML 문서와 함께 처리되므로, 일반 프로그래밍 언어보다 기능이 제한적이다. 자바스크립트는 객체 기반의 언어지만 상속과 클래스라는 개념이 없다. 인터프리터 언어로서 클라이언트의 웹 브라우저에 의해 해석되고 실행된다. 또 HTML에 연산 제어 등 프로그래밍적인 요소를 추가하고 클라이언트의 자원을 활용할 수 있게 한다[23].

2.4.2 Spring Framework

스프링 프레임워크(Spring Framework)는 자바 플랫폼을 위한 오픈소스 애플리케이션 프레임워크로서 간단하게 스프링(Spring)이라고 한다. 동적인 웹 사이트를 개발하기 위한 여러 가지 서비스를 제공하고 있다[24].

스프링 프레임워크는 자바 엔터프라이즈 개발을 편하게 해주는 오픈 소스 경량급 애플리케이션 프레임워크이며, 종속 객체를 생성하고, 조립해주는 도구 역할을 한다. JavaSE로 된 자바 객체(POJO)를 JavaEE에 의존적이지 않게 연결해준다.

자바 객체(POJO)는 오래된 방식의 간단한 자바 오브젝트라는 말로서, Java EE 등의 중량 프레임워크들을 사용하게 되면 해당 프레임워크에 종속된 중량 객체를 만들게 된 것에 반발한 것이다. 주로 특정 자바 모델이나 기능, 프레임워크 등을 따르지 않은 자바 오브젝트를 지칭하는 말로 사용되고 있다[25].

스프링의 특징으로는 그림 12에서와 같이 크기와 부하의 측면에서 경량 프레임워크이며, 제어 역행이라는 기술을 통해 애플리케이션을 결합한다.

또한 간단한 컴포넌트로 복잡한 애플리케이션을 구성하고 설정할 수 있다. 가장 큰 특징으로는 동적 웹 사이트 개발을 위한 대표적인 프레임워크이며, 대한민국 전자정부 표준 프레임워크의 기반 기술이다.

스프링 프레임워크	
모델-뷰-컨트롤러 패턴 지원	자바 객체를 직접 관리하는 경량 컨테이너
POJO 방식의 프레임워크	관점 지향 프로그래밍
제어 반전	의존성 주입

그림 12. 스프링 프레임워크의 특징

2.4.3 Thymeleaf

Thymeleaf는 일종의 자바 라이브러리로, 어플리케이션에 의해 생성된 디스플레이 데이터나 텍스트를 표시하기 위해 템플릿 파일을 변환해 주는 XML 템플릿 엔진이다[26]. Thymeleaf는 웹 애플리케이션에서 사용되는 XML 파일을 모두 처리할 수 있다. 템플릿 파일에 명시적인 로직 코드 대신에, DOM에 사전 정의된 XML 태그와 어트리뷰트를 바탕으로 코드를 작성한다. I/O 연산에 최소한의 자원을 사용할 수 있도록 하여 파싱된 파일의 지능형 캐싱을 통해 빠른 템플릿의 처리가 가능하도록 한다.

Thymeleaf의 핵심은 문서 객체 모델 처리 엔진이다. 특히, Thymeleaf 템플릿 표현 트리를 메모리에 상주시키기 위해 고유의 고성능 DOM을 이용한다. 이 DOM은 노드를 탐색하고 프로세서가 처리될 때 컨텍스트(Context)에 기술된 설정과 데이터셋에 따라 템플릿을 표시하도록 문서 객

체 모델을 수정한다.

웹문서는 오브젝트 트리 구조로 표현되기 때문에 문서객체 모델 템플릿 표현을 사용하는 것은 웹 어플리케이션에 적합하다. 또한 웹 어플리케이션은 많지 않은 수의 템플릿을 이용하는데, 각 템플릿 파일의 크기가 크지 않으며 어플리케이션이 실행될 때는 거의 수정되지 않는다는 점에서 Thymeleaf는 템플릿을 파싱하는 문서객체 모델 트리를 메모리에 상주시키면, 템플릿을 표현할 때 필요에 따라 극히 적은 I/O 처리를 하게 되기 때문에 실행 환경에서 빠른 동작 속도를 보여준다.

2.4.4 MySQL

표준 데이터베이스 질의 언어인 구조화 질의 언어(Structured Query Language)를 사용하는 공개 소스의 관계형 데이터베이스 관리 시스템(RDBMS)이다. 매우 빠르고, 유연하며, 사용하기 쉬운 특징이 있다. 다중 사용자, 다중 스레드를 지원하고, C, C++, Java, PHP, 파이썬(Python) 스크립트 등을 위한 응용 프로그램 인터페이스를 제공한다. 유닉스나 리눅스, 윈도우 운영 체제 등에서 사용할 수 있다. 리눅스 운영 체제와 아파치(Apache) 서버 프로그램, MySQL, PHP 스크립트 언어 구성은 상호 연동이 잘되면서도 공개 소스(오픈 소스)로 개발되는 무료 프로그램이어서 홈페이지나 쇼핑몰 등 일반적인 웹 개발에 널리 이용되고 있다[27].

MySQL은 관계형 데이터베이스 관리 시스템(RDBMS)으로, 데이터베이스를 관리하거나 자료를 관리하기 위한 GUI 관리툴은 내장되어 있지 않다. 따라서 이용자들은 명령 줄 인터페이스 도구들을 이용하거나 데이터베이스 생성, 관리, 데이터 백업, 상태 검사, 데이터베이스 구조를 생성 또는 데이터 레코더를 작성하는데 있어 MySQL 소프트웨어나 웹 애플리케이션을

사용해야 한다[28].

그림 13에서와 같이 MySQL 워크벤치는 사용자에게 MySQL 데이터베이스 관리를 그래픽적으로 지원하게 하며, 데이터베이스 구조의 설계도 시각적으로 하게 해주는 MySQL AB에 의해 개발된 자유로운 집적 환경을 가지고 있다. MySQL 워크벤치는 이용자가 데이터베이스를 설계하고, 모델링, SQL 관리, 데이터베이스 관리까지 할 수 있도록 지원하고 있다.

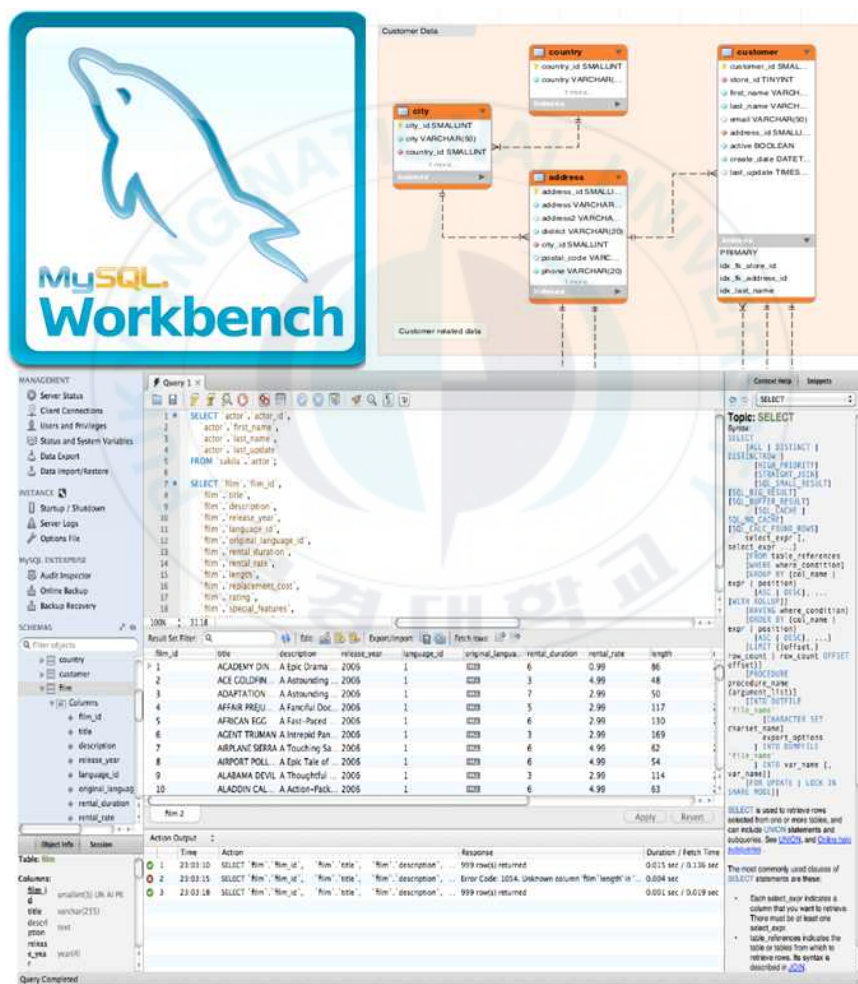


그림 13. MySQL Workbench 프로그램

웹상에서 협업 사용자들 간 데이터를 송,수신하여 데이터를 공유 및 업로드한다. 웹 상에서의 데이터 공유는 제 3자의 침입을 통해 송,수신 데이터를 공격하여 강제로 변형 및 데이터를 패킷 플루딩하여 서버를 마비시킬 수 있다. 이것을 방지하기 위해 메인 작성자가 생성한 웹페이지에 협업 사용자들이 회원가입하여 송수신 데이터를 공유 및 관리하도록 한다[29].



Ⅲ. 제안하는 스토리 작성 협업 시스템

본 논문에서 제안하는 스토리 작성 협업 시스템의 전체 구조를 그림 14와 같다. 제안하는 스토리 작성 협업 시스템은 기존에 사용되고 있는 협업 시스템을 개선하여 고효율적 협업 시스템을 생성하고자 한다. 그 전에 본 논문의 가장 큰 요지인 스토리 작성 협업 시스템의 개념에 대해서 알아본다. 스토리 작성 협업 시스템은 어떠한 구조를 가지고 있는지, 구조에 대한 모듈들은 어떠한 기능을 가지고 시스템이 진행되는지에 대하여 설명한다. 협업형 오픈시나리오 스토리 시스템의 설계와 기능에 대해 기술한다. 마지막으로 스토리 작성 협업 시스템의 전체적인 구조에서 협업 시스템 모듈에서 투표 기반 스토리 작성 협업 시스템의 기능을 설명하고 기존의 협업 시스템의 개선된 부분에 대하여 기술한다.

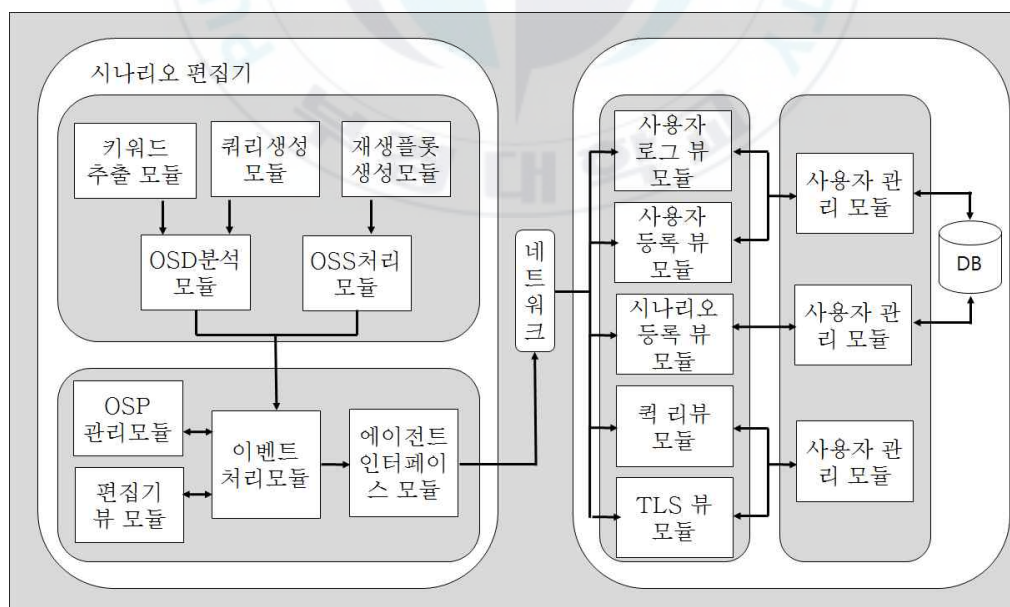


그림 14. 오픈시나리오 스토리 작성 협업 시스템 전체 구조도

3.1 스토리 작성 협업 시스템 개념

스토리 작성 협업 시스템에서 스토리 작성 시스템은 사용자 정의를 지원하는 스토리 작성을 기술된 시나리오이며, 이런 시나리오를 주제로 삼아 여러 사용자들이 협업을 통하여 고품질의 스토리 작성 콘텐츠를 생성할 수 있다.

스토리 작성하는데 있어 시나리오 관련 용어로는 시놉시스(Synopsis)와 트리트먼트(Treatment), 장면(Scene), 샷(Shot), 지문 및 대사가 있다. 시놉시스는 보통 영화의 간략한 줄거리로서, 시나리오의 청사진을 제시할 때 유용하게 사용되는 형식이다. 주제, 기획의도, 등장인물, 줄거리가 시놉시스에 포함된다. 트리트먼트는 장소에 따라 등장인물들과 주요 사건 등을 써놓은 원고이다. 각 장면별로 줄거리를 작성한 것이며, 여기에 대사만 첨가하면 시나리오가 된다. 장면은 시나리오에서 각각의 장면을 일컫는 일종의 가장 기본적인 단락과 같은 단위이다. 주로 장소의 변화나 줄거리의 변화가 발생할 때 장면이 사용된다. 샷은 장면을 구성하는 최소 단위이다. 지문 및 대사는 등장인물의 행동에 대한 객관적인 묘사를 나타낼 때 사용한다. 표 3에서는 시나리오 관련 용어로 제작되는 전통적 영상을 제작하는 단계와 스토리 작성 시스템을 사용하는 영상 제작 단계로 나누었다.

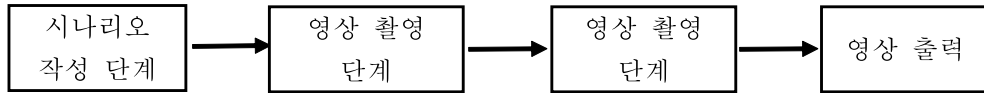
표 3. 전통적 영상 제작 단계와 스토리 작성 시스템 영상 제작 단계

단계	전통적 영상 제작	스토리 작성 시스템의 영상 제작
시나리오 작성단계	장면 별로 시나리오 작성	장면 별로 시나리오 작성
영상촬영 단계	감독이 시나리오에 따라 영상을 촬영하는 단계	작성된 시나리오에 따라 적절한 영상 수집 및 선택
영상편집 단계	촬영한 영상에 장면 전환 등과 같은 편집 기술 적용	수집된 영상에 다양한 변형, 객체, 추가 등 영상을 편집
영상출력 단계	영상 출력	영상 출력

전통적 영상 제작 및 스토리 작성 시스템 영상 제작 단계는 시나리오 작성 단계에서 장면별로 시나리오를 작성하는 것은 동일하지만 시나리오 작성 단계와 영상촬영 단계에서 차이점을 보인다.

그림 15에서와 같이 전통적인 영상 제작 단계에서는 작성된 시나리오를 기반으로 영상을 촬영하고 촬영한 영상을 편집 단계를 거쳐서 영상을 출력한다. 스토리 작성 시스템 영상 제작 단계에서는 작성된 시나리오를 기반으로 인터넷에서 관련 영상을 수집하여 가장 적절한 영상을 선택한다. 영상 편집 단계에서는 수집된 영상들을 음향 효과, 배경 음악, 이미지 등을 삽입하여 다양한 효과를 추가한 이후에 영상을 출력한다. 전통적인 영상 제작에서 편집 단계 진행 중에 시나리오를 수정해야할 경우, 다시 영상을 촬영해야 영상 편집을 할 수 있다는 단점을 가지고 있다. 하지만 스토리 작성 시스템 영상 제작에서는 수집된 영상을 선택하여 영상을 편집하기 때문에 다시 영상을 편집하는 경우가 있다고 해도 수집된 영상들에서 선택하여 편집이 용이하다는 이점을 가지고 있다.

▶ 전통적 영상 제작 단계



▶ 스토리 작성 시스템 영상 제작 단계

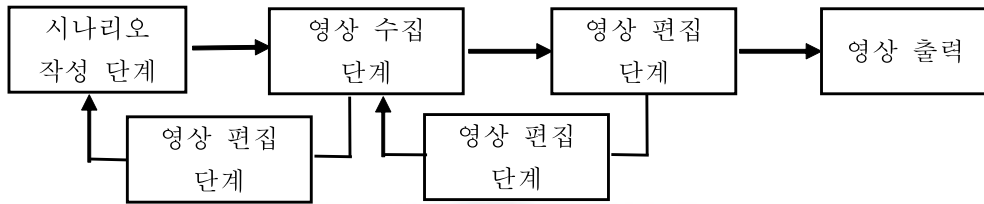


그림 15. 전통적인 영상 제작과 스토리 작성 시스템 영상 제작

표지, 용지, 글자체, 글자크기 등의 전통적인 시나리오 구조를 탈피하여 스토리 작성 시스템에서는 오픈시나리오의 구조인 OSD와 OSS로 모델링한다. 표 4에서는 OSD와 OSS의 구조와 기능을 설명하고 있다.

표 4. OSD와 OSS의 구조와 기능

OSD	· 표준 시나리오 기술자를 정의하여 사용자의 시나리오 창작 활동을 지원 · 이미지, 음향, 배경음악, 자막 등 다양한 객체들을 포함 가능 · Human + Machine Readable 표현 기술 적용
OSS	· 창작된 시나리오를 기반 · 웹 상에 존재하는 다양한 미디어에 대한 재생, 변형, 결합 등을 지원하여 새로운 미디어를 창작할 수 있도록 제어 명령어들을 표현

3.1.1 OSD(OpenScenario Descriptor)

OSD는 시나리오 작성 및 촬영의 관점에서 작성하는 콘텐츠에 대하여 기술하는 내용들이 포함된 것이다. 시나리오의 몇 개의 장면들을 모여 하나의 이야기 단위인 시퀀스를 구성한다. OSD는 그림 16과 같이 카테고리(category), 커버(cover), 시놉시스(synopsis), 시퀀스(sequence), 4가지의 속성으로 구성된다.

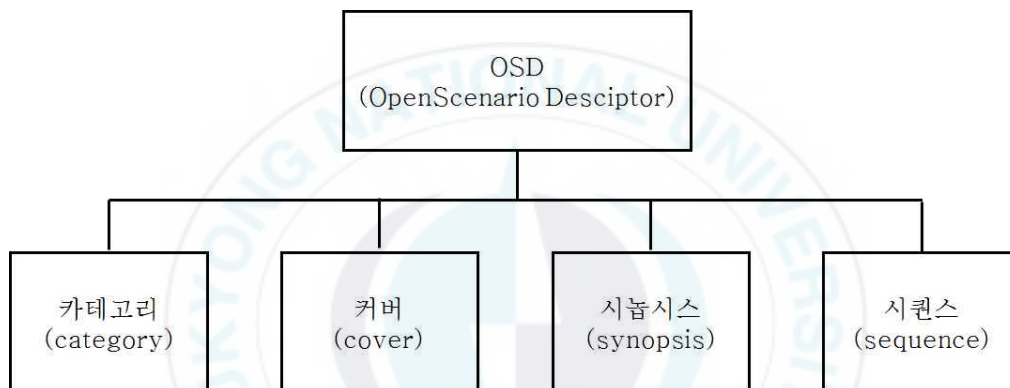


그림 16. OSD의 4가지 속성

카테고리는 스토리 제작 영상의 종류 및 시나리오 모델을 지정한다. 두 가지 속성을 가지는데 도메인(domain)과 모델(model)이다. 도메인은 영상 수집 단계에서의 영상이 포함된 것을 도메인명으로 지정한다. 모델은 두 가지로 구분하는데 BM(Basic Model)과 EM(Extended Model)이 있다.

표 5과 같이 스토리 작성 시스템의 특징 중 하나는 영상 관련 비전문분야 사용자들도 스토리를 작성하여 쉽게 영상을 제작하는 것이다. 영상 관련 업종 종사자들을 위한 확장 모델을 제작하여 효율적인 영상 제작 및 편집하도록 지원한다.

표 5. 기본 모델과 확장 모델 특성

기본 모델 (Basic Model)	• 일반 사용자를 서비스 대상 • 장면별 간단한 문장을 입력하여 영상 제작 • 간단한 영상 편집 기능 제공
확장 모델 (Extended Model)	• 시나리오 작가와 같은 전문가를 서비스 대상 • 전형적 시나리오 구조 지원 • 다양한 영상 편집 기능 제공

커버는 표지를 의미하는 속성을 지니고 있다. 시나리오 제목, 창작자, 연락처 등을 표시한다. 시놉시스는 표 6과 같이 주제(theme), 동기(motif), 등장인물(cast), 줄거리(summary), 4개의 하위 속성으로 구성된다. 하위 속성은 카테고리의 기본 모델(BM)과 확장 모델(EM)과 연계되어 속성 의미가 부여된다.

표 6. 시놉시스 하위 속성

속 성	의 미	모 델
주제(theme)	작사의 의도나 주제를 기술	확장 모델
동기(motif)	시나리오를 작성하게 된 동기를 기술	확장 모델
등장인물(cast)	등장인물을 기술	확장 모델
줄거리(summary)	간단한 줄거리 기술	기본 모델

시퀀스는 장면의 변환과 장면을 관리한다. 장면은 하나 이상의 샷을 표현하는데 사용된다. 장면은 시나리오 번호와 제목, 일어나는 장소와 시간을 의미하는 속성이다.

3.1.2 OSS(OpenScenario Scripts)

OSS는 창작된 시나리오를 기반으로 제작되고, 웹 상에 존재하는 다양한 미디어에 대한 재생, 변형, 결합 등을 지원하여 새로운 미디어를 창작할 수 있도록 제어 명령어들을 표현하는 스크립트 언어다. 고품질 콘텐츠 미디어를 창작하는 여러 가지 기능을 적용하는데 자막 추가, 음향 효과, 배경 음악, 장면의 변형 및 합성 등이 있다.

OSS 기능은 기본 기능(FF, Fundamental Functions)과 확장 기능(EF, Extended Functions)으로 분류된다. 표 7에서는 OSS의 기본 기능과 확장 기능에 대하여 설명한다.

표 7. OSS의 기본 기능 및 확장 기능 특징

기본 기능 (Fundamental Functions)	· OSD를 분석하여 미디어 검색을 위한 키워드 생성 · 생성된 키워드를 이용하여 시멘틱 미디어 검색을 위한 검색 질의어 생성 · 검색된 미디어들로부터 적절한 미디어를 선택 · 시간 정보를 입력하여 미디어 재생 플롯 생성 · 오픈시나리오 파일 포맷 생성
확장 기능 (Extended Functions)	· 미디어의 변형 및 결합 처리 기능 · 객체 기반 제어 기능 · 다양한 미디어 객체 제어 기능 추가

OSS의 특징은 OSD 제어명령과 장면 처리 명령으로 구성된 간단한 구조로 지원이 가능하다. OSD 제어 명령어는 그림 17에서 OSD 처리를 위한 기본 명령어를 수행하고, 키워드 및 검색 질의 생성과 미디어 검색, 재생 플롯을 생성한다.

```
<basic_command> ::= "osdLoad" "<" "osd_load_parameter ">" \
                    "osdAnalysis" "<" "osd_analysis_parameter ">" \
                    "osdQuery" "<" "osd_query_parameter ">" \
                    "osdSearch" "<" "osd_search_parameter ">" \
                    "osdView" "<" "osd_view_parameter ">"
```

그림 17. OSD 제어명령

장면 처리 명령어는 표 8에서와 같이 장면을 처리하기 위한 확장 명령어를 수행하고, 검색된 장면의 객체에 대한 제어 및 처리를 수행한다.

표 8. 장면 처리 명령어

명령어	형식	입력파라미터	비고
장면 객체 선언 및 할당	scene_object_identifier= obj<parameters, [S#]> scene_object_identifier=s cene<S#, obj#>	객체 파일명 장면 번호, 객체번호	엘리먼트 'object'에 추가 또는 검색
장면 객체의 추가	scene_identifier+'scene_ object_identifier		검색된 영상에 object 영상의 추가
장면 객체의 삭제	scene_identifier-'scene_ object_identifier		객체기반의 영상인 경우 object 삭제 기능
장면전환	cutaway<type, S#_1, S#_2>	S#_1, S#_2 : 장면변환 앞뒤 장면번호	

스토리 작성 시스템의 오픈시나리오 전체적인 구조는 그림 18에서와 같이 오픈시나리오 편집기를 이용해 스토리를 작성하여 OSD분석을 통해 OSS 생성 및 실행을 통해 영상을 편집하여 최종 영상을 출력한다.

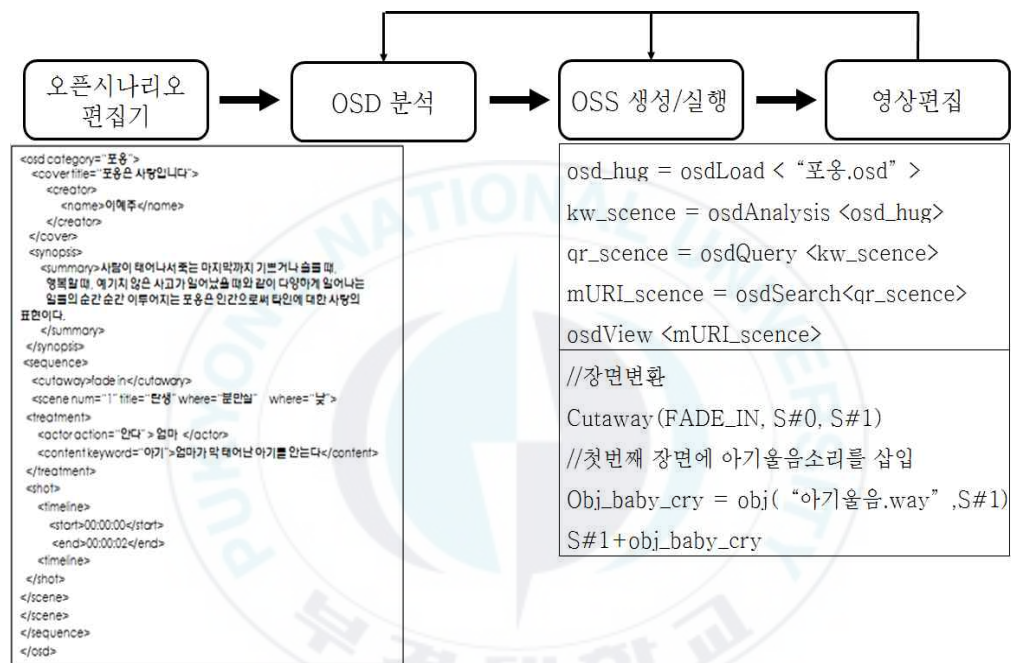


그림 18. 오픈시나리오의 전체적인 영상 편집 구조

3.2 스토리 작성 협업 시스템 구조 설계 및 기능 분석

스토리 작성 협업 시스템의 구조는 오픈시나리오를 입력하고 편집하여 스토리를 제공하는 시스템이다. 사용자가 오픈시나리오를 쉽게 입력할 수 있도록 사용자에게 인터페이스를 제공하고, 입력된 스토리를 분석한다. 콘텐츠 검색을 위한 검색 키워드를 추출하여, 추출된 검색 키워드로부터 검색 질의를 생성한다. 검색된 여러 개의 콘텐츠를 수신하여 적절한 콘텐츠를 선택한다. 선택한 콘텐츠에 사용자가 이미지, 자막과 같은 효과를 추가하여 최종적으로 영상을 재생하고자 하는 재생플롯 정보인 타임라인을 전송하는 기능을 제공하고, 오픈시나리오의 저장, 수정 등의 관리 기능을 제공한다.

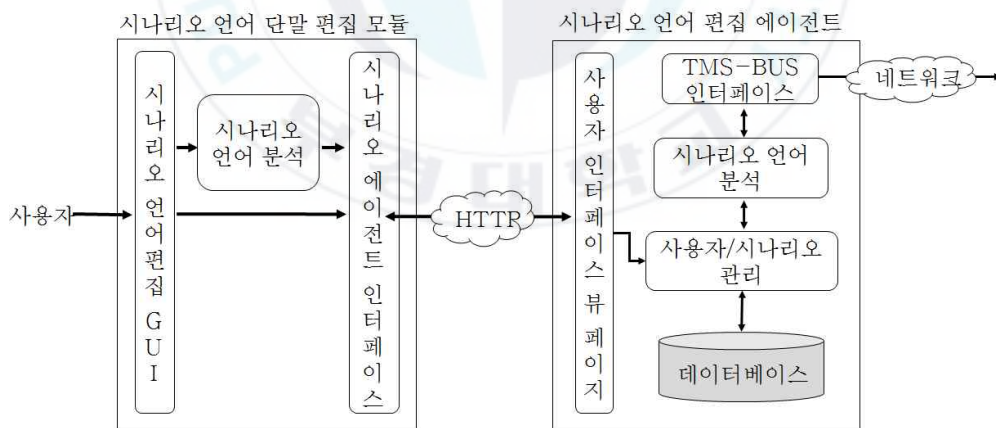


그림 19. 스토리 작성 협업 시스템 구성도

그림 19에서와 같이 스토리 작성 협업 시스템 구성은 시나리오 언어 단말 편집 모듈과 시나리오 언어 편집 에이전트 모듈로 분류된다. 시나리오

언어 단말 편집 모듈은 사용자 단말에 설치된 단말 소프트웨어로써, 시나리오 언어를 작성하고 편집하기 위한 사용자 인터페이스이다. 시나리오 언어 분석 및 시나리오 언어 편집 에이전트 모듈과 인터페이스를 제공한다. 시나리오 언어 편집 에이전트 모듈은 단말과 동일한 시나리오 언어 분석 기능을 제공함과 동시에 시나리오를 관리하고 사용자 간의 시나리오를 공유할 수 있는 서버 기능을 제공한다.

3.2.1 시나리오 언어 단말 편집 모듈

시나리오 언어 단말 편집 모듈의 구조는 그림 20에서와 같이 사용자에게 인터페이스를 제공하는 시나리오 언어 편집 GUI, 시나리오 언어를 분석하는 시나리오 언어 분석, 그리고 네트워크 인터페이스를 제공하는 시나리오 에이전트 인터페이스로 분류한다.

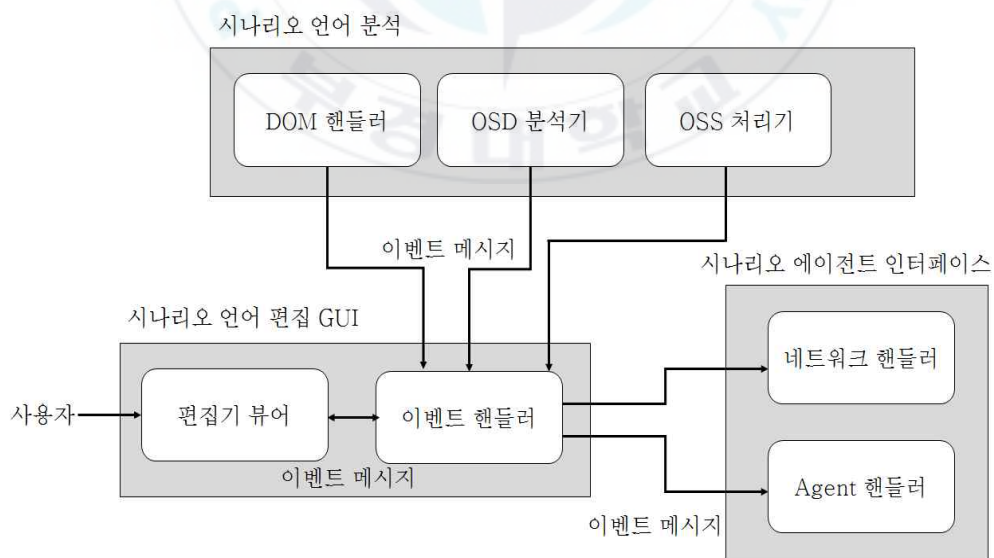


그림 20. 시나리오 언어 단말 편집 모듈 구성도

○ 시나리오 언어 분석 모듈

사용자가 시나리오 GUI 모듈을 이용하여 입력한 스토리를 분석하고 정의된 시나리오 구조화, 분석 및 편집 효과의 추가를 위한 기능을 제공한다. 시나리오로부터 콘텐츠 검색을 위한 검색 키워드를 추출하고 검색 질의문을 생성한다. 검색된 콘텐츠 중 적절한 콘텐츠를 선택한 사용자가 이미지, 자막, 효과음과 같은 효과를 추가하기 위해 입력한 스크립트를 파싱하여 콘텐츠의 재생 플롯 타임라인 정보를 구성한다.

○ 시나리오 언어 편집 GUI 모듈

시나리오 언어 편집 GUI는 사용자가 생성하려는 콘텐츠에 대한 스토리를 입력할 수 있도록 사용자 인터페이스를 제공한다. 또한 입력된 스토리는 처리 단계별 파일로 저장되어 관리한다. 이벤트 핸들러는 사용자에게 따라 시나리오 언어 분석 및 시나리오 에이전트 인터페이스에 대한 이벤트 메시지를 생성하고 해당 모듈을 호출 및 스토리 데이터를 전달한다.

○ 시나리오 에이전트 인터페이스 모듈

시나리오 언어편집 에이전트와의 연결을 통하여 외부 시스템에 검색 질의문과 타임라인 정보를 전송하고, 그 결과를 받아서 시나리오 언어 편집기에서 처리한다. 네트워크 연결과 데이터를 송수신 하는 기능을 제공하며, 시나리오 언어 편집기를 이용하여 사용자가 입력한 시나리오 관리를 위한 요청과 응답을 처리하는 기능을 제공한다.

3.2.2 시나리오 언어 편집 에이전트 모듈

그림 21에서는 사용자 인터페이스를 위한 뷰를 구성하는 사용자 인터페이스 뷰 페이지, 시나리오 언어 편집기와 동일한 기능을 제공하는 시나리오 언어 분석, 사용자 정보 및 시나리오 정보를 처리하고 관리하는 사용자/시나리오 관리, 데이터 전송을 처리하는 TMS-BUS 인터페이스로 분류한다.

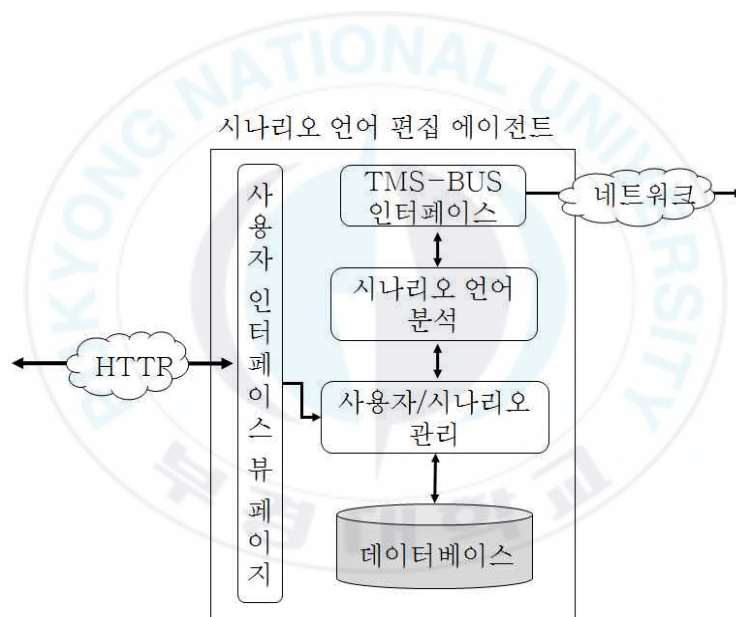


그림 21. 시나리오 언어 편집 에이전트 모듈 구성도

○ 사용자 인터페이스 뷰 페이지 모듈

사용자 인터페이스 뷰 페이지는 편집기의 요청에 따른 사용자 뷰를 제공한다. 사용자 로그인 뷰는 인터페이스에서 사용자가 로그인하기 위한 로그인 요청/응답 기능을 제공한다. 시나리오 관리 뷰는 로그인한 사용자에게 자신이 작성한 시나리오를 관리하기 위한 시나리오 뷰

를 제공한다. 프로젝트 작업공간 생성 요청 기능과 기존에 생성된 시나리오 프로젝트를 수정하기 위해 읽기 요청/응답 기능을 제공한다. 시나리오 언어 입력 뷰는 사용자가 시나리오 언어 편집기에서 작업한 시나리오 언어를 업로드 요청/응답하는 기능을 제공한다.

○ 사용자/시나리오 관리 모듈

사용자/시나리오 관리 모듈은 시나리오 언어 편집 에이전트 모듈을 이용하여 작성된 시나리오를 관리, 공유하기 위해 사용자 및 시나리오를 관리 기능을 처리한다. 사용자 관리는 사용자 로그인 뷰의 요청으로부터 사용자 정보를 얻어 이를 처리하여 데이터베이스에 저장, 수정 및 삭제 등의 기능을 수행하고 사용자 로그인 뷰에 처리결과를 반환한다. 시나리오 관리는 시나리오 관리 뷰에서 요청 받아 시나리오 작업을 생성, 수정, 삭제 등의 관리 기능을 수행하고 처리결과를 시나리오 관리뷰에 반환한다.

○ TMS-BUS 인터페이스 모듈

시나리오 언어 편집기에서 생성한 검색 질의어 및 타임라인 정보를 시나리오 언어 편집 에이전트를 통하여 외부 시스템으로 전송하는 기능을 가진다. TMS메시지 처리는 시나리오 언어 에이전트가 외부 시스템에 데이터를 전달하기 위해 정의된 메시지 형태로 데이터를 변환하는 기능을 제공한다.

3.3 협업형 오픈시나리오 스토리 시스템

스토리 작성 협업 시스템 구성의 사용자/시나리오 관리 모듈에서 사용자와 시나리오를 관리하는 기능 중에 스토리를 작성하기 위해 사용자들 간의 협업하는 시스템 기능을 가진다. 이러한 협업 시스템 기능은 그림 22와 같이 구성된다.

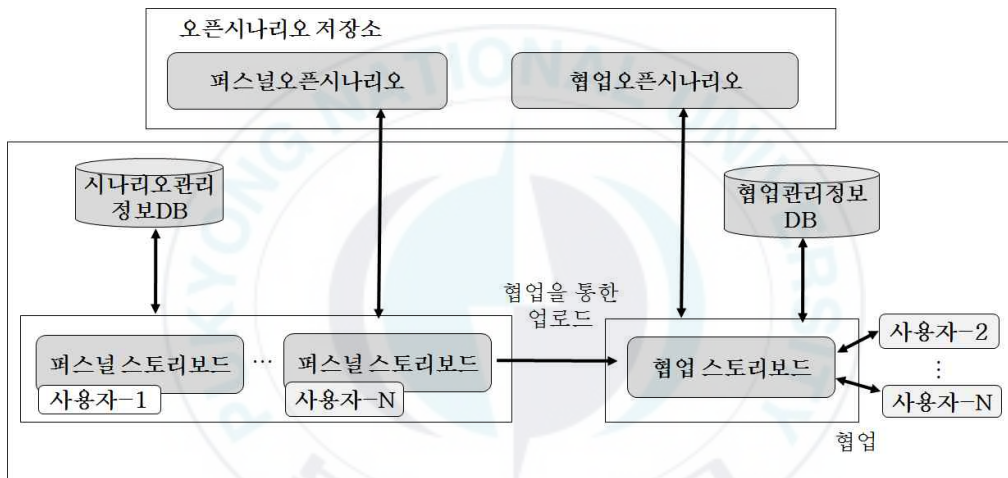


그림 22. 협업형 오픈시나리오 스토리 시스템 전체 구성도

스토리 작성 협업 시스템 구성에서 사용자가 개인만의 스토리 콘텐츠를 원하는 경우에는 퍼스널 스토리보드에 스토리를 작성하여 퍼스널 스토리 시나리오 관리 데이터베이스에 저장하고 오픈시나리오 저장소의 퍼스널오픈시나리오에 작성한 스토리를 저장한다. 개인 지향이 아닌 협업을 통한 스토리 작성을 원하는 사용자는 개인의 퍼스널 스토리보드에 스토리를 작성하고, 작성된 스토리를 협업스토리보드로 업로드한다. 협업스토리보드의 협업관리정보 데이터베이스에는 협업 중인 사용자의 정보와 사용자 개

인 퍼스널 스토리보드에 대한 정보를 저장한다. 오픈시나리오 저장소의 협업오픈시나리오는 협업스토리보드에서 사용자들 간의 협업을 통해 작성된 스토리를 저장소에 저장한다.

협업형 오픈시나리오의 작업 흐름은 크게 사전 작업, 협업 작업 두 가지로 분류한다. 사용자가 협업형 오픈시나리오 생성, 시놉시스 작성, 협업 참여자 등록이 사전 작업에 해당된다. 협업 작업에는 사전 작업으로 구성된 시놉시스와 협업 참여자들이 트리트먼트/신의 내용 작성하고, OSD와 OSS를 작성한다.

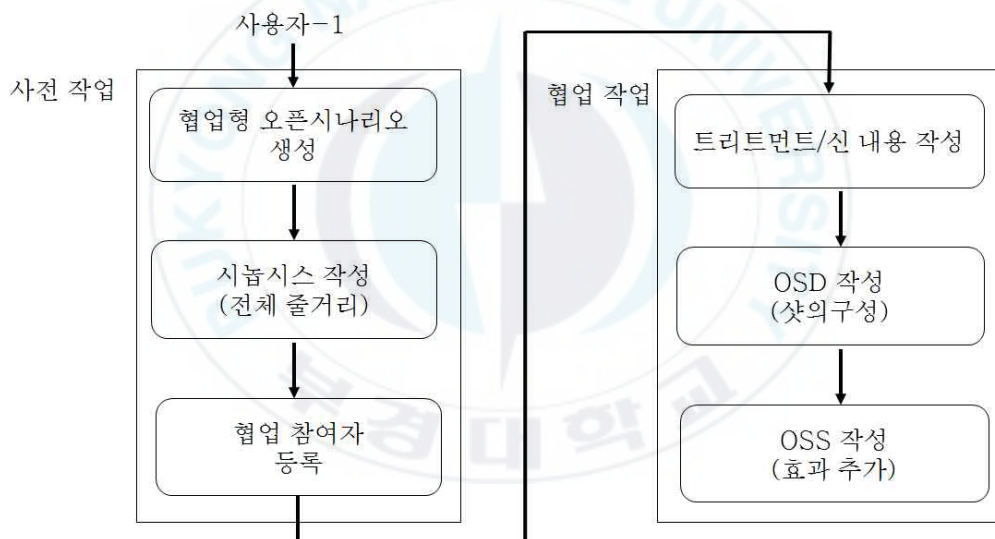


그림 23. 협업형 오픈시나리오 작업 흐름도

협업형 오픈시나리오 시스템의 작업 흐름도는 그림 23과 같으며 그 작업 단계는 다음과 같다. 협업형 오픈시나리오 생성은 사용자-1이 협업형 오픈시나리오로 시나리오 생성 및 등록한다. 사용자-1은 생성된 협업형 시나리오의 메인 작성자로 콘텐츠의 전체 작업을 총괄한다. 시놉시스 작성은 사용자-1이 협업형 시나리오를 작성하기 위해 간략하게 시놉시스를 작성한

다. 협업 참여자 등록은 사용자-1이 협업형 시나리오 작성에 참여할 협업 참여자들을 모집하기 위해 공지하고, 협업 참여자들은 시놉시스를 확인하고 협업 참여자(스토리텔러/편집자)로 등록한다. 트리트먼트/신 내용 작성은 각 협업 스토리텔러들이 전체 줄거리를 기반으로 장면과 샷의 줄거리를 위한 트리트먼트, 장면을 작성한다. OSD 작성에서 각 협업 스토리텔러들은 작성된 트리트먼트를 기반으로 샷의 구성들을 결정하고 샷에 관련된 OSD 내용들을 기술한다. OSS 작성에서 각 편집자들은 이전에 작성된 OSD 내용과 트리트먼트를 확인하고 적용하려는 효과를 설정하여 OSS 내용을 기술한다.



3.4 오픈 시나리오 투표 기반 스토리 작성 협업 기능

현재 사용되고 있는 LMU 방식과 CMM 방식에서 LMU 방식은 협업 사용자들에게 저장소의 접근 권한을 부여함으로써 저장소 사용에 제한을 둔다. 권한 획득 사용자가 읽어온 데이터를 수정하여 업로드해야 권한이 다른 사용자에게 양도되게 된다. CMM 방식은 저장소에 데이터가 업로드될 때 다른 협업 사용자들에게 업로드된 데이터를 복제하여 데이터를 전송하고 다수의 사용자들이 참여 가능한 방식을 사용하고 있다.

하지만 LMU 방식에서 발생하는 문제점의 첫 번째는 사용권한을 지닌 사용자가 데이터를 읽기만 하고 업로드를 하지 않을 경우, 권한이 없는 사용자들은 저장소에 접근할 수 없어 권한 획득 사용자가 데이터를 업로드할 때 까지 무한정 대기 상태인 교착 상태에 빠지게 된다. 두 번째는 협업 사용자들이 각자 다른 장면의 스토리를 작성할 때, 그 전 내용을 확인하지 못하기 때문에 내용의 연계가 자연스럽지 못하다. CMM 방식에서 발생하는 문제점의 첫 번째는 저장소의 원본 데이터에 업로드 데이터가 덮어쓰기 때문에 원본데이터가 보존 되지 않는다. 두 번째는 다수의 협업 사용자들이 CMM 방식을 사용할 경우, 저장소에 협업 사용자들이 데이터를 동시에 업로드 했을 때, 데이터들 간의 충돌이 발생하여 저장소에 저장되는 데이터가 누락되거나 충돌로 훼손된 데이터가 저장소에 업로드 되는 경우가 발생한다.

이러한 문제점을 개선하기 위해 투표 기반의 스토리 작성 협업 시스템을 제안한다. 투표 기반 스토리 작성 협업 시스템은 기존의 파일 공유 방식 문제점들인 교착상태와 원본 데이터 보존 불가 등을 개선하여 여러 협업 사용자들이 스토리를 작성하여 저장소에 읽거나 작성하는데 충돌이 발생하지 않도록 한다.

그림 24에서는 위에 언급한 오픈시나리오 스토리 시스템에서 퍼스널 스토리보드와 협업스토리보드 간의 협업 시스템에서 투표 기반 스토리 작성 협업 시스템을 사용한다.

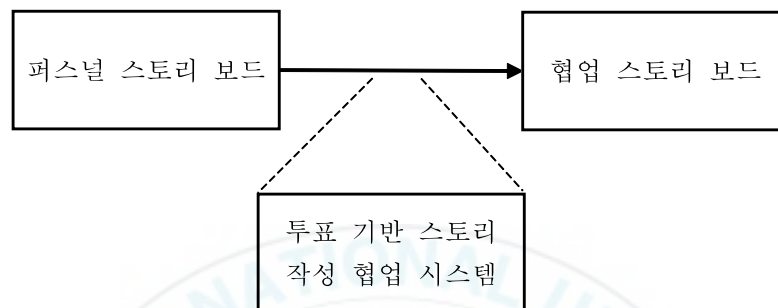


그림 24. 협업형 오픈시나리오 스토리 시스템에서 투표 기반 스토리 작성 협업 시스템

투표 기반 스토리 작성 협업 시스템은 사용자들이 작성한 스토리의 내용을 보고 적합한 스토리를 사용자들의 투표에 의해 결정된다. 투표는 사용자들이 스토리들 간의 내용을 보고 선택하는 말이다.

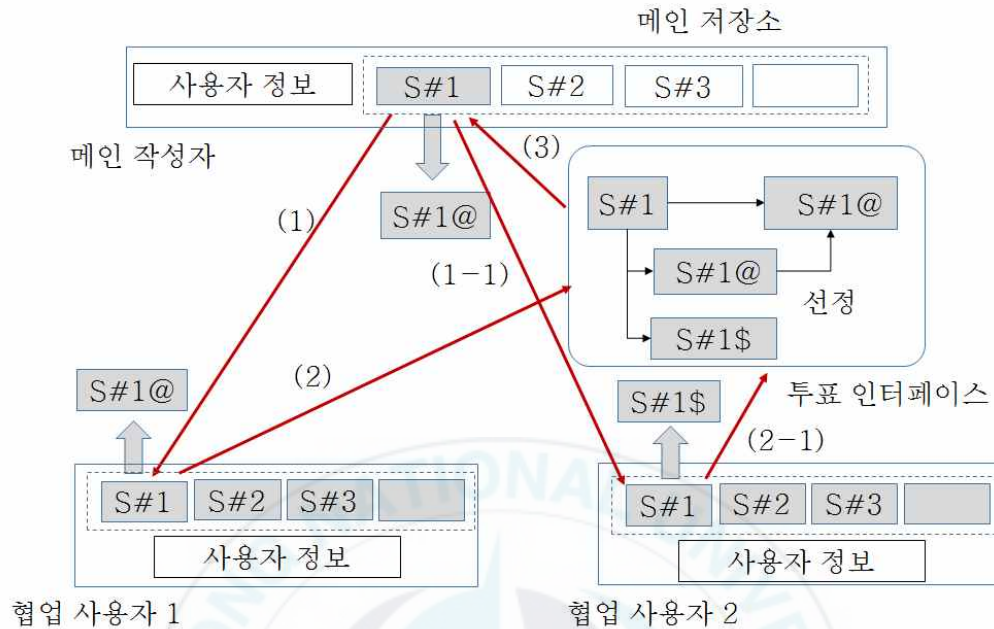


그림 25. 오픈시나리오 스토리 작성의 투표 기반 협업 시스템

투표 기반 스토리 작성 협업 시스템의 처리 과정은 그림 25에서와 같이 시나리오 콘텐츠를 생성한 메인 작성자가 시나리오 콘텐츠의 모든 것을 총괄한다. 협업 참여자로 모인 다른 사용자들은 메인 작성자가 생성한 시나리오 스토리의 메인 저장소에 각 스토리 장면을 선택하여 작성한다. 일반적인 파일 공유 방식은 저장소에 있는 스토리 데이터를 받아 자신만의 스토리로 작성하고 메인 저장소에 업로드하여 스토리 데이터를 수정하지만, 이 스토리 작성 협업 시스템의 투표 공유 방식은 메인 저장소에 업로드하기 전에 투표 인터페이스에 유저들이 작성한 스토리 데이터들을 수집하는 모듈을 생성하여 스토리 데이터들을 수집한다. 투표 인터페이스에 스토리 데이터들이 수집될 때 마다 시나리오 콘텐츠를 이용하고 있는 사용자들에게 투표 참여 메시지를 전송하여 협업 사용자가 투표 인터페이스에 있는 스토리 데이터들을 읽어들 수 있도록 유도한다. 데이터들을 확인한 이후

메인 저장소에 업로드 되는 것에 적합한 스토리 데이터를 투표하여 메인 저장소에 저장된다. 이러한 투표 기반 시스템은 스토리 데이터들 간 충돌을 방지하여 데이터가 누락되거나 훼손되는 것을 방지할 수 있다.

$$\begin{cases} S_j = \max(\frac{\alpha_y}{\alpha_x}), y \leq x \\ S_i = S_j \end{cases} \quad (1)$$

위의 식(1)에서 S_i 은 전체 스토리의 해당 장면의 번호이며, S_j 는 해당 장면에서 협업 사용자들이 작성한 스토리 데이터이며, α_x 는 스토리 작성 협업 시스템에 참여한 사용자의 전체 인원수를 나타내며, α_y 는 참여한 사용자들의 투표수를 나타낸다. 즉, 해당 장면에 참여한 사용자들의 투표를 가장 많이 받은 스토리 데이터가 해당 장면에 채택한다.

오픈시나리오 스토리 작성 투표 기반 협업 시스템에 참여한 사용자들은 집단 지성 단체이므로 정해진 투표 방식뿐만 아니라 다른 투표 방식을 사용하여 해당 장면의 스토리를 지정할 수 있다. 오픈시나리오 스토리 작성 투표 기반 협업 시스템에서 해당 장면 투표 방식, 베이스 스토리 기반 투표 방식, 해당 장면 및 베이스 스토리 기반 투표 방식이 있다.

3.4.1 해당 장면 투표 방식

해당 장면 투표 방식은 각 개별적인 장면에 투표하는 방식으로, S#의 장면에서 다수의 투표를 받은 스토리를 지정하고, 그 다음 S#의 장면에서는 이전 S#의 장면 스토리와 상관없이 해당 장면에서 다수의 투표를 받은 스토리가 지정되는 방식이다. 다음은 각 장면에서 얻은 점수를 계산하는 방

법에 대하여 설명하고자 한다.

$$\begin{aligned} i &= 1, 2, \dots, m \\ \forall i \in \{1, 2, \dots, m\}, \text{score}(i) &= \max(V_i) \end{aligned} \quad (2)$$

식(2)에서 m 는 장면의 총 개수를 나타내고, V_i 는 장면 i 에 대한 사용자의 투표 집합이다. score 는 사용자 투표에 따라 장면 i 에서 얻는 횟수이다. 이 투표 방식을 사용하면 이전 장면의 스토리의 연계를 고려하지 않아 전체 스토리 내용이 매끄럽지 못하다.

3.4.2 베이스 스토리 기반 투표 방식

베이스 스토리 기반 투표 방식은 메인 작성자가 작성한 베이스 스토리를 기반으로 작성한 스토리 데이터를 투표하는 방식은 전체 스토리를 이해하고 있는 메인 작성자가 해당 장면을 분할하여 베이스 스토리를 작성하였으며, 베이스 스토리 기반으로 협업 사용자들이 스토리를 작성하였다.

$$\begin{aligned} i &= 1, 2, \dots, m \\ j &= 1, 2, \dots, n \\ \forall i \in \{1, 2, \dots, m\}, \forall j \in \{1, 2, \dots, n\}, \\ \text{score}(H_{ij}) &= \max(V_{ij}) \end{aligned} \quad (3)$$

식(3)에서는 식(2)와 마찬가지로 m 은 장면의 총 개수를 나타내고, n 은 사용자의 총 인원을 나타낸다. H_{ij} 는 장면 i 의 베이스 스토리를 기반한 사

용자 j 의 스토리 데이터이며, V_{ij} 는 H_{ij} 스토리 데이터에 대한 사용자 투표의 집합이다. 이 방식은 각 개별적인 장면에 투표하는 방식보다 스토리 연계성이 뛰어나다. 하지만 이 방식을 사용하여 작성한 전체 스토리 내용들을 확인했을 때, 스토리 내용의 연계성이 부족한 것을 확인할 수 있다.

3.4.3 해당 장면 및 베이스 스토리 기반 투표 방식

마지막으로 베이스 스토리 기반 및 이전 장면을 인식한 상태로 작성한 스토리 데이터를 투표하는 방식은 두 번째 방식을 보완한 방식이다. 각 장면에 대하여 투표를 가장 많이 받아 메인 저장소에 저장되는 스토리 데이터를 구하는 방법을 설명한다. 식(4)은 식(3)을 기반으로 정의한다.

$$\begin{aligned} & \forall_i \in \{1, 2, \dots, m\}, \forall_j \in \{1, 2, \dots, n\} \\ & \text{if } i = 1, W_1 = \max(H_{1j}) \\ & \text{if } i > 1, W_i = C(W_{i-1}, H_{ij}) \end{aligned} \quad (4)$$

여기서, W_i 는 각 장면 i 에서 가장 투표를 많이 받은 스토리 데이터이다. C 는 이전 스토리(W_{i-1})의 내용과 가장 자연스럽게 연결되는 현재 작성하려는 스토리(H_{ij})를 찾는다. 그리고 스토리 내용의 연계는 각 사용자의 판단으로 투표를 통해 결정된다. 메인 작성자가 작성한 베이스 스토리를 기반으로 작성된 스토리 기반과 이전 장면의 스토리 내용을 이해한 협업 사용자들이 작성한 스토리를 투표를 통해 적절한 스토리 데이터를 선정한다. 이 방식은 스토리 내용의 연계성이 3가지 투표 방식 중 가장 뛰어나다. 지정된 방식으로 행동하는 프로그램이 아닌 각각의 자율적인 지성을 가진 집

단 지성체인 사람과 사람간의 협업 시스템이기 때문에 협업 사용자들이 어떤 투표 방식을 사용할지는 알 수 없다. 하지만 투표 방식의 효율성을 이해한다면 가장 효율적인 투표 방식을 채택할 것이다.

오픈시나리오 스토리 작성 협업 시스템에 참여하기 위해서는 회원가입을 해야한다. 따라서 데이터의 보안성에 대해서는 메인 작성자가 회원가입을 한 협업 사용자에게만 데이터를 송/수신하기 때문에 보안성이 용이하다.

만일 투표의 수가 동일한 스토리 데이터가 발생할 경우, 하나의 장면 메인 저장소에 투표 수가 동일한 갯수의 스토리 데이터들이 업로드 된다. 그렇게 된다면 CMM의 문제점인 데이터 간의 충돌로 인한 누락 및 훼손이 발생하게 된다. 이러한 문제점을 해결하기 위해 투표 수가 동일한 스토리 데이터들 중에 해당 저장소에 적절한 스토리 데이터를 메인 작성자의 선택하여 동일한 투표 수로 인해 발생하는 문제점을 방지할 수 있다.

IV. 실험 결과 및 고찰

제안한 투표 기반 작성 협업 시스템을 검증을 위하여, 기존 드라마 시나리오를 활용하여 협업 시스템을 구현하였다. 앞에서 설명하였듯이, 현재 사용되고 있는 협업 시스템 서비스에서 협업을 통해 시나리오를 제작하는데 문제점을 해결하기 위하여 오픈시나리오 협업 기능에 투표 기법을 추가하였다. 사용자들이 생성한 스토리가 메인 저장소에 즉각적으로 저장되는 것이 아닌 투표를 통하여 일반 사용자들도 정해진 장면에 적절한 스토리를 지정하여 메인 저장소에 저장되도록 하였다. 그리고 실험 도구로는 Java Eclipse 프로그램을 이용한 제안기법을 구현하였고, MySQL을 활용하여 웹을 통하여 생성된 스토리를 데이터를 관리하는 데이터베이스를 생성하였다.

그림 26은 드라마 시나리오를 이용하여 협업 시스템을 구현하였다. 협업 시스템의 메인 화면인 Dashboard에서는 드라마의 제목, 작가, 메인, 줄거리를 생성하여 일반사용자들에게도 어떤 스토리를 생성할지를 보여준다. 관리자 관점의 메인화면과 일반 사용자 관점의 메인화면 차이점은 메인화면에서 Write 버튼의 유무이다. 관리자에게 Write 버튼을 추가함으로써 장면을 생성하는 권한을 가진다. 그림 26에서 S#6와 S#7는 메인화면에서 관리자가 Write 버튼으로 생성한 장면의 스토리이다. 관리자인 메인 작성자가 Dashboard에서 S#을 생성할 시, 전체적인 시나리오에서 나누어지는 S#들로 구성된다.

그림 27은 Dashboard에 구현된 데이터들을 MySQL 데이터베이스에서 관리한다.

Dashboard

Write

태양의 후예

박상현

Story

낮선 땅 극한의 환경 속에서 사랑과 성공을 꿈꾸는 젊은 군인과 의사들을 통해 삶의 가치를 담아낼 블록버스터급 휴먼 멜로 드라마

Character

유시진

강모연

서대영

윤영주

S#1_A

3

군사정보지휘통제실 (밤)

김철수

S#2_A

3

비무장지대 경계초소 안 (밤)

김철수

S#3_B

3

비무장지대 경계초소 주변 (밤)

이미나

S#4_C

3

군사정보지휘통제실 (밤)

나현우

S#5_A

3

비무장지대 경계초소 주변, 울밭 앞 맥북지 (밤)

김철수

S#6

0

비무장지대 나무 위 (밤)

박상현

S#7

0

비무장지대 울밭 앞 맥북지 (밤)

박상현

Dashboard

태양의 후예

박상현

Story

낮선 땅 극한의 환경 속에서 사랑과 성공을 꿈꾸는 젊은 군인과 의사들을 통해 삶의 가치를 담아낼 블록버스터급 휴먼 멜로 드라마

Character

유시진

강모연

서대영

윤영주

S#1_A

3

군사정보지휘통제실 (밤)

김철수

S#2_A

3

비무장지대 경계초소 안 (밤)

김철수

S#3_B

3

비무장지대 경계초소 주변 (밤)

이미나

S#4_C

3

군사정보지휘통제실 (밤)

나현우

S#5_A

3

비무장지대 경계초소 주변, 울밭 앞 맥북지 (밤)

김철수

S#6

0

비무장지대 나무 위 (밤)

박상현

S#7

0

비무장지대 울밭 앞 맥북지 (밤)

박상현

그림 26. 관리자모드에서의 협업 시스템 메인화면과 일반 사용자모드에서의 협업 시스템 메인화면

seq	scena	title	content	write	like_count
6	S#1	군사령부 지휘통제실 (범)	중앙 스크린에는 경계초소의 위성사진이 떠 있고, 북한군을 나타내는 붉은 점 3개와 국군을 나타내는 파란점 2개가 보인다. 작전장교 : 현...	박상현	0
7	S#2	비무장지대 경계초소 안(범)	북한 특작부대원 2는 무장해제 당한 국군병사 둘(병장 1, 일병 1)에게 총을 겨누고 있다. 다른 북한군 특작부대원 1은 창밖을 향해 총을 겨...	박상현	0
8	S#3	비무장지대 경계초소 주변 (범)	초소 주변에 매복하여 포위 중인 수색대대 병사들.	박상현	0
9	S#4	군사령부 지휘통제실 (범)	을글중 중장 : 북한군의 목적은 우리가 먼저 공격하는 것을 기다리고 있습니다. 미한수 수색 : 그러면 어떡합니까? 먼저 공격하면 전쟁이...	박상현	0
10	S#5	비무장지대 경계초소 주변 풀밭 앞 매복지 (범)	대위계급 중대장 옆의 수색대 병사가 풀밭을 향해 총구를 겨누 채 수색대 1 : 정지! 움직이면 쏜다! 풀밭에서 시진, 대영, 공하사가 나타난...	박상현	0
11	S#6	비무장지대 나무 위 (범)	임종사 : (무전)피클로 송신 완료, 목표 위치 확보했습니다. 조준경을 초소에 있는 북한군을 향해 겨눈다.	박상현	0
12	S#7	비무장지대 풀밭 앞 매복지 (범)	매복지의 중대장과 수색대대 병사들은 시진과 대영을 보고 있는데, 뒤에서 최종사가 나타난다. 최종사 : (무전)스누피 송신 완료. 폭발물...	박상현	0

그림 27. MySQL을 이용한 Database 기능 구현

그림 28에서 S# number를 클릭하게 되면 Dashboard에서 Scenario View로 넘어간다. Scenario View는 메인 작성자가 S# number마다 관련된 베이스 스토리를 작성한다. S# number, S# Title, 작성자, S# Text로 Scenario View를 나타내고 있다. 작성된 S#의 베이스 스토리가 메인 스토리의 가장 큰 디자인을 가지게 되며, 협업 파트너인 일반 사용자들은 베이스 스토리를 확인한다. 이때, Scenario View에서도 Write 버튼을 클릭하여 일반 사용자들이 베이스 스토리를 바탕으로 자신만의 스토리를 작성할 수 있는 Story Write를 구현하였다. 메인 작성자 및 일반 사용자들로 인해 생성된 스토리는 S# Text 밑에 구현하도록 하였다.

그림 29에서는 Scenario View에 구현된 스토리 데이터와 작성자 등의 정보들을 MySQL 데이터베이스에서 저장 및 관리한다.

Scenario View
Write

S#1	군사령부 지휘통제실 (밤)	박상현
-----	----------------	-----

중앙 스크린에는 경계초소의 위성사진이 떠 있고, 북한군을 나타내는 붉은 점 3개와 국군을 나타내는 파란점 2개가 보인다

작전장교: 현재 상황에 대하여 언급

지휘 통제실에서는 장성 윤길중 중장과 청와대 외교안보수석 이한수가 상황보고를 받음

이한수 수석: 도대체 북한군과 뭘 하려고 저기 있는 겁니까?

윤길중 중장: 북한이 우리에게 도발을 하고 있는 겁니다

이한수 수석: 도발요? 전쟁이라도 하겠어요?

S#1
0
군사령부 지휘통제실 (밤)
박상현

S#1_A
3
군사령부 지휘통제실 (밤)
김철수

S#1_B
1
군사령부 지휘통제실 (밤)
이미나

S#1_C
1
군사령부 지휘통제실 (밤)
나현우

그림 28. Dashboard의 S# number의 Scenario View 화면

Result Grid	Filter Rows	Edit	Export/Imports	Wrap Cell Contents																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																											
-------------	-------------	------	----------------	--------------------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

그림 29. Database의 Scenario View 데이터

Story Write

S#2_A	군사정부지휘통제실(밤)	user1
-------	--------------	-------

20대 초반의 옛된 북한군 전사2가 무장해제 당한 채 포로로 잡힌 국군 병사들(병장 1, 일병 1)에게 총을 겨누고 있다. 양 쪽 모두 긴장한 모습. 이에 비해 상대적으로 고참으로 보이는 전사 1은 창밖을 향해 총을 겨누고 있고, 그 옆에는 눈에서 관자놀이까지 깊게 베인 칼자국이 있는 (자략)북한군 안정준 상위가 초코파이를 까먹으며 무심한 표정으로밖을 쳐다본다.

Save

Cancel

그림 30. Scenario View의 Story Write 화면

그림 30의 Scenario View에서 메인 작성자와 일반 사용자들이 작성한 스토리들을 클릭하여 Story View로 화면이 변환된다. Story View는 Scenario View에서 베이스 스토리를 확인하고 사용자들이 자신만의 스토리를 작성하여 보여주는 화면이다.

S# number에서 구분을 짓기 위해 뒤에 알파벳 순서로 표기하였고, 베이스 스토리를 필연성을 가지고 스토리를 작성하지 않기 위해 S# Title은 수정 가능하도록 설정하였다. Story View에서 작성자가 아닌 다른 사용자들은 읽기만 가능하도록 설정하였고, 작성자 본인은 작성한 스토리에 대하여 삭제, 수정 할 수 있도록 버튼을 추가하였다.

Story View

S#2_A	비무장지대 경계초소 안 (밤)	김철수
-------	------------------	-----

20대 초반의 앳된 북한군 전사2가 무장해제 당한 채 포로로 잡힌 국군병사 둘(병장 1, 일병 1)에게 총을 겨누고 있다. 양 쪽 모두 긴장한 모습. 이에 비해 상대적으로 고함으로 보이는 전사 1은 창박을 향해 총을 겨누고 있고, 그 옆에는 눈에서 관자놀이까지 깊게 베인 칼자국이 있는 (자락)북한군 안정준 상위가 초코파이를 까먹으며 무심한 표정으로 밖을 쳐다본다.

Vote

3

Delete

Modify

Cancel

그림 31. Scenario View의 Story View 화면

그림 31에서 보여주는 투표 기법을 활용하여 Vote 버튼은 사용자들이 현재 S# number에서 적절하다고 판단하다고 생각하면 투표하기 위한 버튼이다. Vote의 수치는 Scenario View에서 일반사용자가 작성한 스토리들에 표시되며, 투표의 수치가 가장 높은 S# 스토리가 해당 S# Number에 적절한 스토리로 판단하여 Dashboard에 구현되도록 하였다.

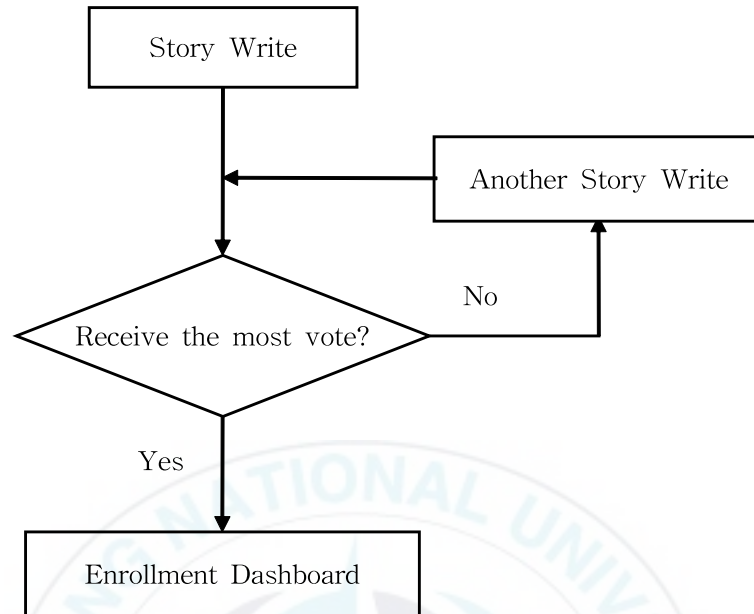


그림 32. 제안한 기법 실험의 투표 알고리즘

그림 32는 제안한 투표 기법의 실험 알고리즘이다. 작성된 스토리 데이터에 대하여 가장 투표를 많이 받았는지를 확인하고, 투표를 가장 받지 않을 경우 선택된 스토리가 아닌 다른 스토리를 선택하여 가장 투표를 많이 받았는지를 확인한다. 투표를 가장 많이 받은 스토리 데이터는 Dashboard에 등록된다.

그림 33은 프로그램을 투표 기법 알고리즘의 진행과정을 나타낸다. Scenario View에서 협업 사용자들이 작성한 스토리를 확인하여 해당 장면에 적절한 스토리 데이터를 투표한다. 투표의 수를 가장 많이 받은 스토리 데이터가 자동으로 Dashboard에 있는 해당 장면의 메인으로 등록된다.

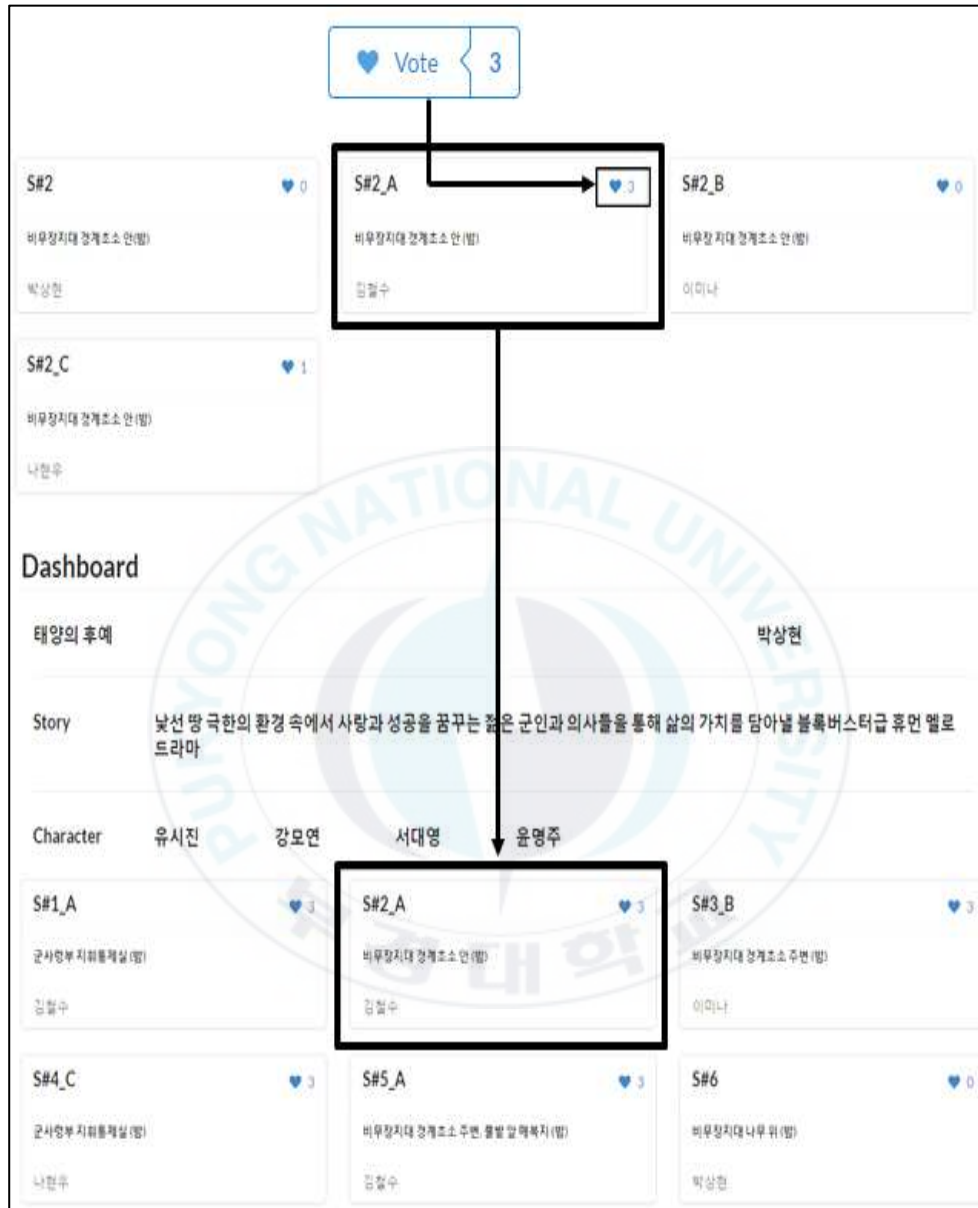


그림 33. 투표를 가장 많이 받은 스토리가 Dashboard에 등록

제안한 기법을 이용한 실험을 통하여 표 9에서와 같이 기존의 협업 시스템 서비스인 아마존닷컴 스토리 라이터와 구글 스토리빌더를 비교하여 기존의 협업 시스템 서비스의 문제점들을 보완한 것을 확인하고 표 10에서 투표 기법 협업 시스템을 사용했을 때 기존의 협업 서비스의 문제점들을 개선한 것을 확인할 수 있었다.

표 9. 아마존닷컴 스토리 라이터와 구글 스토리빌더 문제점

아마존닷컴 스토리 라이터	• 스토리를 작성하는데 복잡한 사용방법 • 협업 작업 시, 작성한 스토리 파일을 협업 사용자의 이메일에 송신하는 번거로운 파일 공유 방식 • 협업 사용자 간의 각자 작성한 스토리 파일을 확인하는데 있어 불필요한 시간소요
구글 스토리 빌더	• 실시간 채팅기반 스토리 작성 프로그램이므로 협업 사용자가 자리를 비웠을 경우, 작업 진행이 불가 • Lock and Unlock 방식을 채택하므로 우선순위 사용자가 작업을 협업 사용자에게 권한을 넘기지 않을 경우, 우선순위가 아닌 다른 사용자들은 교착상태에 걸림 • 웹 상의 링크를 통해서만 스토리 작성 과정 동영상으로 업로드가 될 뿐, 작성한 스토리 파일을 다른 협업 사용자들에게 이메일 통해 전달하지 못함

표 10. 제안한 기법을 통해 기존의 협업 시스템 서비스의 문제점 개선

오픈시나리오 스토리 작성 투표 기반 협업 시스템	• 간편한 플랫폼과 사용방법 • 사용자들 간 작성하고자하는 스토리에 대한 웹 상에서 협업 작업 도모 • 문서 작업 기반의 스토리 작성 프로그램으로 협업 사용자들의 불필요한 시간에 맞추어 참여 강요가 불필요 • 투표 방식의 사용으로 메인 저장소에 작성된 스토리 데이터들 간의 충돌 회피, 데이터의 누락을 방지
---	--



V. 결 론

본 논문에서는 스토리 작성 콘텐츠를 생성하고자 하는 사용자들의 요구를 충족시킬 수 있는 투표 기반 스토리 작성 협업 시스템 기법을 제안하였다. 제안 기법은 스토리 작성 협업 콘텐츠를 생성하는데 있어 기존의 협업 시스템 서비스인 아마존닷컴 스토리 라이터와 구글 스토리빌더의 문제점들을 개선하는 새로운 플랫폼을 구현하였다. 아마존닷컴 스토리 라이터의 문제점들은 스토리를 작성하는데 있어 서식을 사용하는 방법이 복잡하며, 협업 작업할 때 스토리 데이터를 작성할 때마다 협업 사용자의 이메일에 송신하는 번거로운 진행 방식을 사용하고 있다. 또 협업 사용자 간의 각자 작성한 스토리 데이터를 확인 및 수정하여 다른 협업 사용자에게 전달하는데 불필요한 시간을 소요하는 경우가 발생한다. 구글 스토리 빌더의 문제점들은 실시간 채팅 기반 프로그램이기 때문에 스토리를 작성할 때 다른 협업 사용자가 자리를 비우거나 존재하지 않을 경우, 스토리 작성이 불가능하다. 스토리빌더는 Lock and Unlock 파일 공유 방식을 사용하기 때문에 우선순위 사용자가 스토리 작성하는데 있어 다른 협업 사용자에게 권한을 이양하지 않으면 우선순위가 아닌 다른 협업 사용자들은 무한정 대기인 교착상태에 빠져들게 된다. 스토리 작성 작업 진행을 하고 다른 협업 사용자에게 작업 데이터를 공유할 때, 웹 상에서 링크를 통해서만 스토리 작성 진행 과정 동영상으로 업로드가 될 뿐, 작성한 스토리 데이터 파일을 다른 협업 사용자들에게 직접적으로 전달하지 못한다.

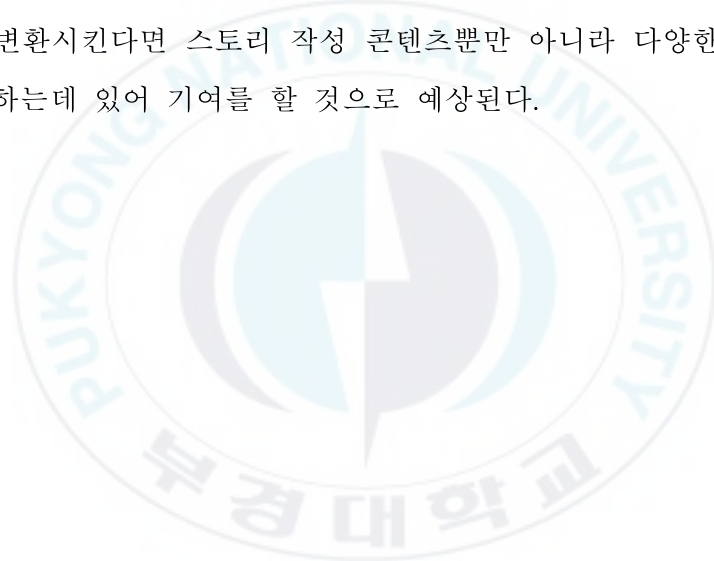
이러한 기존의 협업 시스템 서비스들의 문제점을 개선하고 스토리 작성 콘텐츠를 생성하기 위해 제안한 투표 기반 스토리 작성 협업 시스템은 웹

상에서 스토리 작성 콘텐츠의 생성자, 즉 메인 작성자가 특정 시나리오에 대한 스토리 작성하기 위해 콘텐츠를 생성한다. 로그인 화면에서 메인 작성자의 플랫폼과 일반 사용자의 플랫폼을 구분하였다. 메인 작성자 플랫폼은 전체 시나리오의 장면들을 보여주는 Dashboard를 구현하여 전체 시나리오의 스토리를 작성하기 위한 S#을 생성할 수 있도록 한다. 일반 사용자 플랫폼에서는 전체 시나리오 S#들을 생성할 수 없도록 설정하였다. 메인 작성자는 협업하기 위한 시나리오 S#들의 베이스 스토리를 작성한다. 협업 사용자들은 작성된 각 S#의 베이스 스토리를 확인하고 사용자 개인의 스토리를 작성하여 업로드한다. 업로드된 스토리 데이터들은 Scenario View에 저장되어 메인 작성자와 일반 사용자들 모두가 확인할 수 있도록 구현하였다. 작성된 스토리들 중에서 해당 S#에 적절하다고 판단되는 스토리에 투표하여 각 S#에서 투표를 가장 많이 받은 스토리가 Dashboard에 등록되도록 구현하였다.

제안한 투표 기반 스토리 작성 협업 시스템은 위에서 언급한 기존의 협업 시스템 서비스들의 문제점들을 개선하였다. 간편한 플랫폼과 사용방법을 통해 스토리를 작성하는데 불편했던 점을 보완하였고, 사용자들 간 작성하려는 스토리에 대하여 작성한 스토리 파일 데이터를 일일이 협업 사용자들에게 이메일을 통해 보내는 비효율적인 파일 공유 방식을 웹 상에서 협업을 진행하여 해결하였다. 채팅 기반이 아닌 문서 작업 기반의 인터페이스로 생성하여 다른 협업 사용자들과 협업 작업에 대한 약속 시간을 정할 필요가 없어졌다. 마지막으로 투표 방식의 사용으로 메인 저장소에 작성되는 스토리 데이터들 간의 충돌을 회피하여 스토리 데이터가 누락되는 것을 방지하였다.

실험 결과에서 보듯이, 기존의 협업 시스템 서비스의 문제점들을 우수하게 보완하고 생성하여 진행하기 어려웠던 스토리 작성 협업 콘텐츠를 효율

적인 방면에서 생성할 수 있었다. 하지만 스토리 작성만을 고려한 제안한 협업 시스템 기법으로 고품질의 콘텐츠를 생성할 수 있다는 점은 확신할 수 없다. 이에 대하여 고품질의 콘텐츠를 생성하기 위한 여러 방법들을 지속적인 연구가 필요하다. 마지막으로, 제안한 협업 시스템 기법을 통하여 스토리 작성에 관련된 문서 작업 이외에도 이미지, 자막, 효과음과 같은 효과를 추가하여 고품질의 스토리 작성 협업 콘텐츠를 생성하기 위해서는 협업 시스템에 대한 지속적인 연구가 필요로 한다. 또 기존의 협업 시스템 서비스에 제안한 협업 시스템 기법을 부합시켜 스토리 작성에 알맞은 인터페이스로 변환시킨다면 스토리 작성 콘텐츠뿐만 아니라 다양한 협업 콘텐츠를 생성하는데 있어 기여를 할 것으로 예상된다.



[참고 문헌]

- [1] M. Chen, G. Eason, B. Noble, and I. N. Sneddon, “Big Data: A Survey,” *Mobile Networks and Applications*, Vol. 99, Issue 2, pp. 171-209, April 2014.
- [2] 박대서, 안두현, 홍성은, 김화중, “빅데이터 공유 활성화를 위한 시스템의 제안,” *한국통신학회*, Vol. 60, Issue 6, pp. 1207-1208, 2016년 6월.
- [3] 박상현, 이혜주, 권기룡, 이석환, 박윤경, 문경덕, “콘텐츠 제작을 위한 협업 DB 시스템의 구조 설계,” *한국멀티미디어학회논문지*, Vol. 18, Issue 2, pp. 476-477, 2015년 11월.
- [4] R. Perry, G. Belrose, S. Spence, E. Simpson, M. Rhodes, D. Banks, J. Erickson, and J. Rutherford, “Content-Centered Collaboration Spaces in the Cloud,” *IEEE Internet Computing*, Vol. 13, Issue 5, pp. 34-42, Sept. 2009.
- [5] 박지수, 김무철, 노승민, “멀티미디어 콘텐츠의 맞춤형 정보 제공 연구,” *한국전자거래학회지*, Vol. 20, Issue 3, pp. 79-87, 2015년 5월.
- [6] W. H. Dutton, “The Wisdom of Collaborative Network Organizations: Capturing the Value of Networked Individuals,” *Prometheus*, Vol. 26, Issue 3, pp. 211-230, Sept. 2008.
- [7] 이혜주, 박상현, 권기룡, “사용자 정의 스토리 작성을 위한 사용자간 협업 처리에 관한 분석,” *한국멀티미디어학회논문지*, Vol. 19, Issue 2, pp.400, 2016년 10월.
- [8] D. Banks, J. S. Erickson, and M. Rhodes, “Toward Cloud-based Collaboration Services,” *HotCloud*, June 2009.

- [9] Amazon.com Story Writer, <https://studios.amazon.com/help/amazon-story-writer>, Accessed to 2016.
- [10] Google Story Builder, <https://docsstorybuilder.appspot.com/>, Accessed to 2016.
- [11] R. Drfail, D. Harkey, and J. Edwards, *The Essential Client/Server Survival Guide*, WILEY, Jan. 1999.
- [12] 이해주, 권기룡, 이석환, 박윤경, 문경덕, “사용자 정의 스토리 기반 콘텐츠 제작 서비스를 위한 오픈시나리오 언어 구조 설계,” *한국멀티미디어학회논문지*, Vol. 19, Issue 2, pp. 170-179, 2016년 2월.
- [13] How to Write a Screenplay : Script Example & Screenwriting Tips Writer \Store, <http://www.writersstor.com/how-to-write-a-screenplay-a-guide-to-scriptwriting/>, Accessed to July 2016.
- [14] B. Collins-Sussman, B. W. Fitzpatrick, and C. M. Pilato, “Version Control with Subversion: For Subversion 1.7: (Compiled from r4515),” *O'Reilly Media*, Copyright 2008.
- [15] 박상현, 이해주, 이석환, 권기룡, 박윤경, 문경덕, “클라우드 기반 협업 콘텐츠 서비스 침입탐지 프레임워크,” *한국멀티미디어학회논문지*, Vol. 19, Issue 1, pp. 307-309, 2016년 5월.
- [16] 신성문, “현금흐름 할인법(DCF)를 이용한 Amazon.com 기업가치 평가,” *정보통신연구진흥원*, Vol. 12, Issue 4, pp. 56-62, 2000년 3월.
- [17] 박상현, 이해주, 이석환, 권기룡, 박윤경, 문경덕, “콘텐츠 제작을 위한 데이터 저장용의 새로운 저장소 플랫폼,” *한국멀티미디어학회논문지*, Vol. 19, Issue 2, pp. 290-291, 2016년 10월.
- [18] 이해주, 권기룡, 이석환, 박윤경, 문경덕, “스토리 기반 콘텐츠 제작을 위한 스토리 편집기 설계,” *한국멀티미디어학회논문지*, Vol. 18, Issue

- 2, pp. 681-683, 2015년 11월.
- [19] Writing Scenario, <http://terms.naver.com/list.nhn?cid=42219&category=51139>, Accessed to July 2016.
- [20] H. J. Lee, K. R. Kwon, S. H. Lee, Y. K. Park, and K. D. Moon, "Analysis of Scenario and Story Representational Structure for Story-based Content Creation," *Proceeding of Korea Information Processing Society*, Vol. 22, Issue 2, pp. 1553-1554, Oct. 2015.
- [21] H. J. Lee, S. H. Park, K. R. Kwon, S. H. Lee, Y. K. Park, and K. D. Moon, "Collaborative OpenScenario Platform for Story-based Content Creation," *Proceeding of Multimedia Information Technology and Applications*, Vol. 12, Issue 7, pp. 16-17, July 2015.
- [22] Java Script, <http://webdeveloperskorea.com/archives/134>, Accessed to 2016.
- [23] 조민호, 김동원, "웹에서의 STEP 데이터 공유," *한국경영과학회*, Vol. 21, Issue 1, pp. 1-5, 1998년 1월.
- [24] F. Gutierrez, *Introducing Spring Framework : A Primer*, Apress, June 2014.
- [25] Spring Framework POJO, https://ko.wikipedia.org/wiki/Plain_Old_Java_Object Accessed to 2016.
- [26] Thymeleaf, <http://www.thymeleaf.org/> Accessed to 2016.
- [27] M. McLaughlin, *MySQL Workbench : Data Modeling & Development*, McGraw-Hill Osborne Media, April 2013.
- [28] U. Shokhrukh and J. I. Kim, "A Virtual Clipboard for Instant Data Sharing in a Multi-Device Environment," *Korea Computer Graphics Society*, Vol. 6, Issue 7, pp. 150-151, July 2016.

- [29] 박광용, 송유진, “ID 기반 암호방식을 이용한 데이터 공유와 권한 관리,” *정보보호학회지*, Vol. 19, Issue 4, pp. 78-84, 2009년 8월.

