The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)
Vol. 17, No. 3, pp.275-282, Jun. 30, 2017. pISSN 2289-0238, eISSN 2289-0246

https://doi.org/10.7236/JIIBC.2017.17.3.275
JIIBC 2017-3-33

BIM 기반의 협력적인 건축 설계를 위한 Version Control 시스템

Version Control System for BIM-based Collaborative Architectural Design

배홍민*, 김병서**, 정재희***

Hong-Min Bae*, Byung-Seo Kim, Jae-Hee Chung*****

요 약 차세대 3차원 건축설계 기법으로 각광받고 있는 BIM(Building Information Modeling)기술을 기반으로 하는 건축 설계 프로그램 중 대표적인 프로그램으로 Revit 프로그램이 있다. Revit은 정보 입력 측면을 강화시켜 설계 단계에서부터 구성요소의 정보 패밀리라는 오브젝트로 만들어 알기 쉽게 구성 요소를 파악 할 수 있는 프로그램이나, 이 프로그램에서 추출된 도면 출력파일들을 쉽게 비교하는 프로그램이 전무하고, 이로 인하여 협업 시에 수정된 동일 도면들에 대한 변경 내용의 분석이나 도면 정보의 손쉬운 공유가 어려운 점이 문제로 대두되고 있다. 이에 본 논문에서는 Revit을 통한 출력 도면 파일들의 병렬적 협력 작업이 효율적으로 이루어지도록 하기 위하여 동일 도면에 출력 파일들에 대한 버전 관리 시스템의 제안 및 구현을 하였으며 다른 버전 도면 파일들 간의 수정 부분을 검출하기 위한 개선된 KMP알고리듬을 제안한다.

Abstract Revit program is one of the architectural design programs based on the BIM (Building Information Modeling) technology, which is attracting attention as a next-generation three-dimensional architectural design technique. Revit is a program that enhances the information input side and makes it easy to understand components by making it an object called an information family of components from the design stage, but there is no program that easily compares the drawing output files extracted from this program. It is difficult to analyze the changes to the same drawings modified at the time of collaboration and easily share the drawing information. In this paper, we propose and implement a version management system for the output files on the same drawing, so that the parallel collaboration work of output drawing files through Revit can be efficiently performed. We propose an improved KMP algorithm.

Key Words : IT Service, Version Control, BIM

Ⅰ. 서 론

최근 건축 분야에서는 BIM(Building Information Modeling)을 이용한 건축 설계가 대형 건물의 설계를 위한 차세대 건축설계 방법으로 각광받고 있다.[1] BIM은 건축물의 기획, 설계, 시공, 유지관리의 모든 단계에 필요

*정회원, Nikken Sekkei in Japan IT
**종신회원, 컴퓨터정보통신공학과
***정회원, 홍익대학교 건축공학부
접수일자: 2017년 3월 28일, 수정완료: 2017년 4월 28일
게재확정일자: 2017년 6월 9일

Received: 28 March, 2017 / Revised: 28 April, 2017 / Accepted: 9 June, 2017
**Corresponding Author: jsnbs@hongik.ac.kr
Dept. of Computer and Information Communication Eng., Hongik University, Korea

한 물리적 형상, 속성 및 관련 정보를 호환가능하고 재사용가능한 방법으로 생성하고 관리하는 도구이다.[2] BIM을 적용한 설계 프로젝트는 초기 상태부터 형상화나 물량에 대한 정확한 정보 덕분에 프로젝트를 참여하는 모든 사람들의 의사를 정확하게 전달할 수 있도록 지원한다. 또한 협업을 할 때 설계 단계부터 협업을 진행 할 수 있으므로 전체적인 설계 과정에서 비용 절감의 효과도 볼 수 있게 된다.[3] 위와 같이 BIM기술은 설계 단계에서부터 구성 요소를 쉽게 파악할 수 있기 때문에 설계 단계에서의 비효율 적인 부분의 감소 및 여러 사람이 협업을 할 때 협업작업을 실시간으로 공유하여 설계 프로젝트를 수정하거나 빠르게 정보를 공유 할 수 있다는 장점이 존재한다. 또한 각기 다른 분야(건축, IT, 전기 등)에서 같은 도면을 가지고 자신의 분야의 작업을 바로 공유 할 수 있다. 또한 이러한 정보들이 다른 사람들과 실시간으로 공유되기 때문에 기존의 단계적으로 진행되는 작업과정의 비효율을 줄일 수 있다.

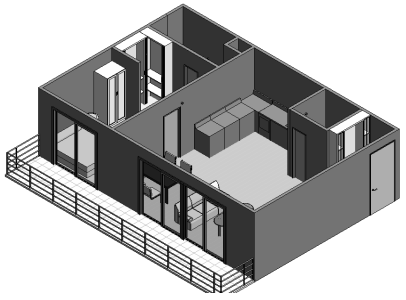


그림 1. Revit을 이용한 공간설계의 예
Fig. 1. Example of spatial design using Revit

이러한 BIM의 개념을 실질적으로 건축설계에 이용하도록 도와주는 프로그램들 중 대표적인 프로그램으로 오토데스크사가 개발한 Revit 프로그램을 들 수 있는데 그림 1에서 Revit을 활용한 BIM 기반의 공간 설계의 예를 보이고 있다. BIM은 건물의 3차원 모델 속에 시공, 설계, 유지관리를 통틀어 설계사가 건축물의 생성부터 철거에 이르기까지 모든 정보를 쉽게 파악하고 건축물에 대한 보완을 할 수 있는 기법이기 때문에 Revit 상에서도 도면을 점과 선으로 나타내는 것이 아닌 객체들의 집합으로 나타내어 설계가 이루어지며 그 도면의 출력 파일은 방대한 양의 객체 데이터를 가지게 된다. 이러한 방대한 데이터를 가진 설계 파일에 대한 공동 작업의 경우 여러 설계자들에 의한 변경 및 수정 사항들에 대하여 이를 추적하거나 제어할 필요가 발생하게 된다. 이에 본 논문에서는 Revit의 출력 도면 파일들을 활용한 공동작업의 경우에 효과적으로 수정된 파일들에 대한 내용을 관리하기 위한 버전 제어 시스템을 제안하고 구현한다. 이 시스템에서는 수정 이전 버전의 데이터와 이후의 데이터 파일간의 신속한 데이터 비교를 위하여 기존의 KMP 알고리즘을 개선한 알고리즘 또한 제안 및 구현하였다.

논문에서는 먼저 2장에서 연구 동기에 대해 설명하고, 3장에서는 Version Control과 문자열 검색 알고리즘에 대해서 소개한다. 이 후 4장에서 협업을 위한 Version Contrl 프로그램 설계에 대한 프로그램을 구현하고자 한다.

II. 연구 동기

Revit을 사용하게 되면 오브젝트 개념인 패밀리를 활용함으로 건축설계 과정 중 개념 설계부분에서의 효율을 최적화할 수 있고, 간단한 모델링도 가능하며 협업 시에 패밀리 단위로 정보를 공유하고 설계 도면을 파악할 수 있다. Revit에서 패밀리(Family)란 여러 가지 부분의 건축 설계부분에 들어가는 구성 요소(예를 들어 벽, 지붕, IT센서 등)들을 객체화 시킨 데이터로서 Revit 프로그램 자체에서 사용자가 보기에 알기 쉬운 이미지를 통하여 건축설계에 있어 위와 같은 구성 요소들에 대하여 설계자가 정의하면 패밀리라는 파일로 Revit내에 저장이 된다.

위와 같은 패밀리는 단순한 객체로 존재하는 것은 아니다. 패밀리의 속성의 크기와 내용은 설계자에 따라서 수정되거나 삭제 될 수 있다. 또한 패밀리에서 이어지는 모든 것을 포함한다. 예를 들자면 "창문" 이라하면 실생활에서는 고유한 의미를 가지는 형태와 크기를 갖는 단 하나의 객체이지만 Revit 에서 "창문" 패밀리는 고유한 창의 속성과 같은 여러 가지 재질이나 크기, 구현 방법까지 포함한다.

이러한 Revit을 통한 패밀리의 출력 파일은 도면의 크기에 따라 막대한 양의 데이터를 포함하게 되는데 이 데이터에 대한 내용은 Revit 프로그램을 사용하여야만 이해될 수 있는 형태의 출력파일로 생성되고 이로 인하여 동종의 도면 설계도나 패밀리에 대하여 공동 작업을 수행시에 각 설계자들에 의하여 수정된 내용을 확인하기 어렵고 수정사항에 대한 트래킹이 어려워 사실적으로 병렬적인 공동 작업보다는 순차적인 작업이나 같은 도면으

로 공동작업이 아닌 새로운 작업을 수행 하는 등의 비효율성이 존재하여 왔다.[4] 이러한 업무의 비효율성을 해결하기 위하여 Revit상에서 작성된 설계도면 및 패밀리의 데이터 출력을 Excel 파일의 형태로 바꾸기 위한 한 선행 연구가 진행되었다.[5] 즉 기존의 Revit의 설계 도면의 출력 파일들을 Excel 파일 형태로 전환함으로써 Revit의 프로그램을 수행하지 않아도 설계 도면상의 요소들에 대한 정보를 쉽게 공유할 수 있으며 수정/가공 또한 가능하게 되었다.

그럼에도 불구하고 다수의 설계자가 병행적으로 같은 도면에 대한 수정 및 가공 그리고 그러한 도면의 공유를 효율적으로 혼동없이 운용하기 위해서는 수정된 내용과 파일에 대한 제어 및 트래킹이 필요할 수 밖에 없다. 이러한 문제를 해결하기 위하여 본 논문에서는 Revit 프로그램의 Ecxel형태의 출력파일을 효과적으로 제어 및 트래킹하기 위한 시스템과 기존 문서와 수정 문서와의 내용 비교를 좀 더 효율적으로 수행할 수 있는 비교 알고리즘을 제안하고자 한다.

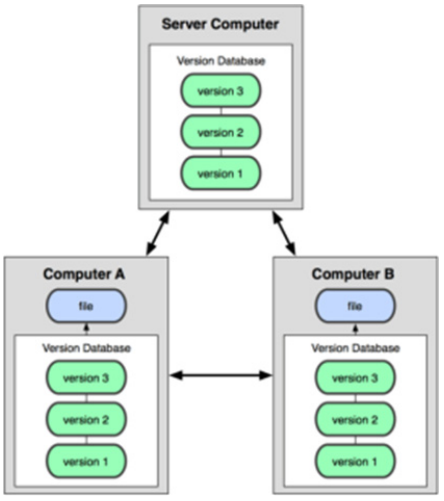


그림 2 중앙 서버형 Version Control Sytem
Fig. 2. Central Server Type Version Control System

Ⅲ. 엑셀파일 기반 Version Control 시스템

1 Version Control 시스템

구현하고자 하는 버전관리 시스템은 Revit 프로그램의 출력을 엑셀파일화 하는 선행 연구[5]의 결과를 활용하여 엑셀 기반의 패밀리 설계파일들에 대한 버전관리시스템이다. 일반적인 건축 설계의 공동작업시에는 설계반장이 존재하기 때문에 서버 하나를 두고 파일을 교환하는 그림 2와 같은 서버형 버전 관리 시스템을 채택하였다. 서버에 모든 파일을 업로드 하여 접근 할 수 있기 때문에 여러 사람이 접근하여 사용하는 구조에서 이점을 볼 수 있다.

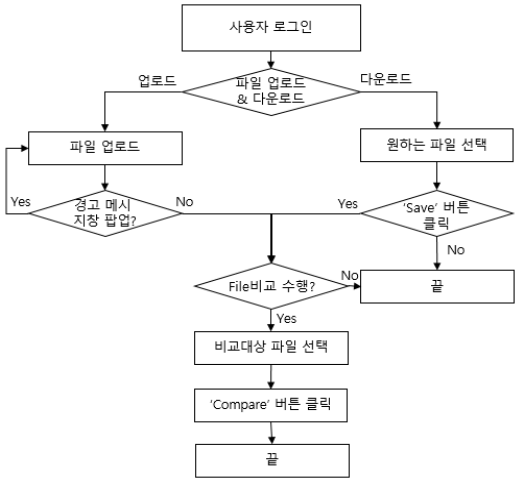


그림 3. 기본적인 프로그램 사용 절차
Fig. 3. Basic program use procedure.

서버 즉 관리자는 여러 설계자 별로 아이디를 생성하여 프로젝트를 관리하게 된다. 이때 프로젝트를 관리하는 방법은 관리자가 프로젝트 별로 폴더를 생성하고 각각의 폴더가 프로젝트 파일의 저장소가 된다. 같은 프로젝트에 포함되어진 사용자들을 그룹으로 묶어 지정된 폴더에만 접속 할 수 있도록 설정한다. 마지막으로 프로젝트 파일의 저장소에는 관리자가 설정한 이름 이외에 파일은 업로드 되지 않도록 설정한다. 위와 같은 설정을 통하여 사용자의 계정에 따라 서로 다른 저장소를 가지게 되며 관리자의 설정에 따른 파일이름의 버전 관리만을 허용하게 된다.

사용자는 다음절에서 설명되는 제안하는 클라이언트 프로그램을 통하여 아래의 사항들에 대한 작업을 수행할 수 있다.

- 기존의 패밀리 파일을 다운로드 및 Revit 프로그램없이도 속성들을 확인/수정/가공
- 설계자의 수정된 패밀리 파일과 기존 버전의 동일 패밀리 파일에 대한 비교 및 변경 사항 확인

- 수정된 설계자의 패밀리 파일 업로드시 버전을 나타내는 파일 이름으로 그 업로드 파일 이름의 자동 변경

본 시스템에서 관리자에 의하여 최초에 올려진 파일을 제외하고 그 이후의 수정된 버전들에 대하여 아래와 같은 형태의 파일 이름의 포맷을 사용한다.

$$\text{파일이름}(\text{사용자}ID)\text{월}-\text{일}-\text{시}-\text{분}.xlsx$$

이러한 포맷을 사용하여 어느 설계자가 어느 시점에 수정본을 올렸는지를 확인할 수 있도록 하였으며, 앞서 언급한 바와 같이 설계자가 시스템에 수정한 자신의 파일을 올리면 자동으로 위와 같은 형태로 파일이름이 변경되어 저장되어진다.

그림 3은 사용자 측면에서의 본 제안 시스템에 대한 사용 절차에 대해서 나타내고 있다. 먼저 사용자가 로그인을 하게 되면 파일을 서버로 업로드 할 것인지 서버에 있는 Excel파일을 다운로드 할 것인지 선택을 하게 된다. 먼저 파일을 업로드 할 때에는 'Upload'버튼을 사용하여 업로드를 진행한다. 이때 지정된 파일이름이외의 파일을 선택하였을 경우에 경고 메시지창이 팝업되며 더 이상 절차가 진행되지 않는다. 이때 지정된 파일 이름은 서버에 생성되는 (파일이름)(사용자ID)(생성날짜).xls에서 파일이름 란의 파일 이름이 프로젝트에서 사용하는 파일이름과 같다면 같은 파일로 인식한다.

다음으로 다운로드만 사용할 때에는 원하는 파일을 선택하고 'Save' 버튼을 사용하여 내가 원하는 로컬 장소에 파일을 저장할 수 있도록 구현하였다.

마지막으로 이전 파일과 비교검증이 필요하다면 서버 파일 리스트 창에서 파일을 선택하면 첫 번째 파일 View 칸에 파일이 열리게 되고 'Compare'버튼을 클릭하면 두 파일간의 다른 부분이 노란색으로 음영되어 나타나게 된다. 이와 같은 동작 시나리오 기반의 시스템 구현을 4장에서 자세히 설명하고 있다.

2 KMP기반 엑셀 파일 비교 알고리즘

2.1 KMP 문자열 검색 알고리즘

버전 관리에서 가장 중요한 기능 중에 하나는 이전 파일과 어느 부분이 수정되어 졌는지 비교하는 기능이다. 이러한 기능을 사용하기 위해서 문자열에 대한 검색 알고리즘을 사용하여 수정된 부분을 찾을 수 있는 구조를 구축 하였다.

문자열 검색 알고리즘이란 문자열 중에서 패턴을 찾아내어 같은 문자열을 찾는 알고리즘으로 대표 적으로 Native String Search, Finite-state automaton based search, Knuth-Morris-Pratt(KMP) Algorithm, Rabin-Karp string search algorithm 등이 존재한다.

KMP 알고리즘은 접두사 함수(prefix function)를 사용하여 문자열이 존재하는 범위를 검색함으로써, Naïve 알고리즘에서 필요 없는 비교 연산을 줄이도록 한 알고리즘이다.[6]

이들 중 KMP 알고리즘은 Kunth, Morris, Pratt 3명의 이름을 따서 KMP알고리즘 이라고 불린다. 문자열 패턴 매칭은 입력으로서 기존의 파일인 텍스트 T 와 입력 받은 패턴 P 가 주어진며, 둘은 문자열이고, 프로그램에서는 문자의 배열로 표현되어 검색을 하게 된다. 즉, 기존파일인 텍스트에서 패턴이 나타나는 모든 위치를 찾아내게 되는 것이다. 예를 들어 텍스트 abcdef 라고 가정한다고, 패턴이 def라고 한다면 패턴이 네 번재와 여섯 번째로 나타남을 알 수 있게 된다. 그림 4는 KMP알고리즘의 개념 코드로써[7] KMP알고리즘의 시간 복잡도는 O(T+P)로 표현되어진다.

```
KMP-MATCHER(T, P)
1   n = T.length
2   m = P.length
3   π = COMPUTE-PREFIX-FUNCTION(P)
4   q = 0                                  // number of characters matched
5   for i = 1 to n                         // scan the text from left to right
6       while q > 0 and P[q + 1] ≠ T[i]
7           q = π[q]                       // next character does not match
8       if P[q + 1] == T[i]
9           q = q + 1                      // next character matches
10      if q == m                          // is all of P matched?
11          print "Pattern occurs with shift" i − m
12          q = π[q]                       // look for the next match
```

그림 4. KMP알고리즘의 개념 코드[7]
Fig. 4. KMP algorithm concept code

2.2 제안하는 Excel 파일 기반 KMP 알고리즘 (E-KMP)

본 시스템에서 버전 관리를 위한 설계파일은 엑셀의 형태를 취하고 있고 이러한 설계도면 Excel 파일들의 비교를 위하여 기존의 KMP알고리즘을 변형한 E-KMP 알고리즘을 제안하고 이를 시스템에 반영하였다. 엑셀파일의 셀기반 데이터 표현방식을 활용하여 좀 더 효율적으로 비교가 가능하도옥 한 KMP 알고리즘을 에서는 파일정보에 맞도록 수정된 KMP알고리즘을 사용하여 원하는 문자열의 정보를 찾는다. 오토마타와 비슷한 형식을 따

르나 상태전이함수가 훨씬 간결하고, 준비 과정도 선형 시간인 점을 생각하면, 문자의 종류가 다양한 상황에서라면 당연히 KMP를 선택해야 메모리와 시간 양측에서 이득을 볼 수 있다. 파일의 정보가 꽤 많은 정보를 포함하고 있고, 각각의 명칭이 중복되지 않기 때문에 KMP알고리즘을 베이스로 하여 수정하였다.

```
KMP-MATCHER(T,P)
n = T.length
m = P.length
π = COMPUTE-PREFIX-FUNCTION(P)
q = 0
for i = 1 to n
        while q > 0 and P[q+1]≠T[i]
                q = π[q]
        if P[q+1] == T[6byte]
                q = q + 1
        if q ==m
                print "Pattern occurs with shift" i – m
                q = π[q]
```

그림 5. 프로그램에 맞도록 수정된 E-KMP 알고리즘
Fig. 5. E-KMP algorithm modified for the program

하지만 기존의 KMP알고리즘에서는 모든 문자열 또는 문장들에 대하여 특정 패턴을 비교 검토해야 하기에 설계 패밀리의 엑셀 파일과 같이 방대한 설계 데이터를 포함하고 있는 경우에는 다소 비교시간이 오래걸리는 문제가 있다. 그러나 일반적인 텍스트 형태의 데이터와는 달리 본 시스템에서 다루고자 하는 설계 패밀리 파일은 Excel파일의 형태를 이루고 있고 엑셀 파일과 같은 포맷은 셀 단위로 단어를 포함하고 있고 설계파일을 Excel로 만들 때 문장 단위로 출력이 되지 않아 모든 패턴을 찾을 필요도 없다. 즉 셀단위로 첫 번째 단어를 비교함으로써 그 셀에 대하여 다음 문자를 비교할 필요가 있는지 없는지를 파악할 수 잇으며 불필요시에는 다음 셀로 진행을 이어나아감으로써 파일 비교 수행에서 좀 더 신속성을 추구 할 수 있다. 이에 KMP알고리즘을 수정하여 셀 단위로 검색을 진행하고 불필요한 문자열을 사전에 파악하여 검색하지 않도록 조정 하였다. 기존이 KMP 알고리즘에서 가장 최악의 문자열 패턴 검색은 TxP번 진행되는 과정이다. 이때 T는 문자열 P는 패턴이므로 이 과정에서 T번의 과정 중에서 찾지 않아도 되는 많은 부분을 제거해 줄 수 있다면 좀 더 빠르게 데이터를 찾을 수 있게 된다. 그림 5와 같이 파일을 비교할 때 기존의 모든 파일을 검색하는 것을 특정 행과 특정 열로 제안해서 찾을 수 있도록 하고, 패턴을 찾을 때에 처음 패턴이 맞는 부분만을 검색하여 검색을 최대한 줄일 수 있도록 하였다. 처음 패턴이 맞는 부분만 찾는 이유는 Revit에서 엑셀파일을 꺼낼 때에 정보가 앞의 패턴이 맞지 않으면 데이터의 종류가 다른 점을 이용한 것이다.

2.3 E-KMP 알고리즘에 대한 성능 평가

제안되어진 E-KMP 알고리즘의 성능을 검증하기 위하여 설계 패밀리 엑셀 파일들을 E-KMP와 기존의 KMP알고리즘들을 사용하여 수행 시간을 평가하였다. 본 실험을 위하여 비교 검색 대상 파일들은 Revit에서 작성된 도면 출력 Excel 파일들을 대상으로 하였으며 임의로 그 엑셀 파일을 수정하여 부분적인 정보가 상이한 도면 파일을 만들어 이 두 파일, 즉 원본 도면 파일과 임으로 수정된 도면 파일들간의 문자열 검색 시간을 파일의 사이즈를 달리 해가며 평가 하였으며 실험 대상 엑셀 파일들은 10, 20, 50 Mbyte의 세가지 형태의 사이즈로 생성하였고 각각에 대하여 50번씩 알고리즘을 수행하여 그 값을 평균하였다. 이렇게 실험한 성능을 그림 5에서 보이고 있다.

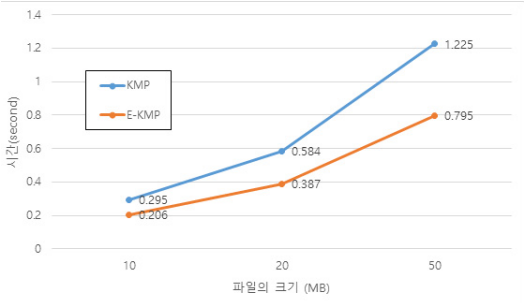


그림 6. 검색 파일 크기에 따른 KMP 알고리즘과 E-KMP 알고리즘의 검색시간 비교 결과
Fig. 6. Comparison of search time of KMP algorithm and E-KMP algorithm according to search file size

그림 6에서 보이는 바와 같이 E-KMP의 검색시간이 짧은 것을 볼 수 있는데 이는 도면 파일들을 검색할 때 KMP 알고리즘의 경우에 모든 데이터를 검색하고 패턴을 전부 찾아야 끝나는 것과 달리 E-KMP알고리즘은 앞에서 일치 하지 않는 부분에 대해서는 끝까지 패턴을 검색하지 않기 때문에 효율성 적인 측면에서 많은 차이가 나게 된다. 또한 파일의 크기가 커질수록 차이가 점점 증가 하는 것을 그림 5를 통하여 볼 수 있다.

Ⅳ. 개발 프로그램의 동작 시현 및 출력 검증

1 제안 시스템의 구현

앞 장에서 설명되어진 버전 관리를 위한 시스템은 C# 언어를 사용하여 클라이언트를 구현하였으며 Database를 Ubunto OS 기반의 서버에 구현하였다.

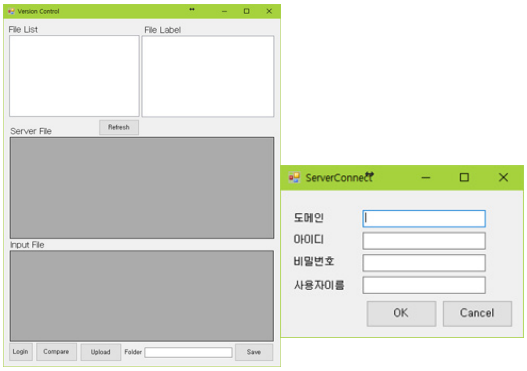


그림 7. 개발 프로그램의 메인 Interface(좌)와 로그인 창(우)
Fig. 7. The main interface (left) and login window (right)

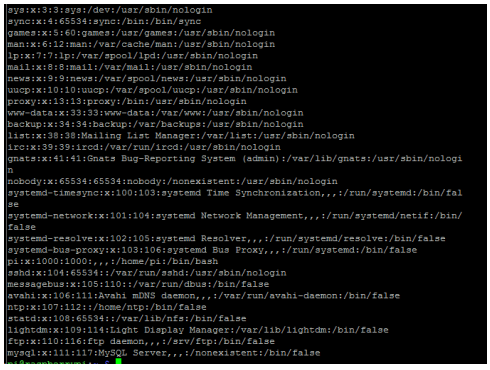


그림 8. 서버에서의 사용자 생성 프로그램 부분
Fig. 8. Program Part showing User Creation on the Server

그림 7 (좌)는 버전 컨트롤을 위한 클라이언트 프로그램의 메인 인터페이스로써 로그인 버튼을 누르게 되면 그림 7 (우)의 창이 띄워지고 서버의 도메인을 입력하고, 아이디와 비밀번호, 사용자 이름을 기입하면 정상적으로 로그인이 되었다는 알림과 함께 서버에 접속할 수 있게 된다.

그림 8은 서버에서의 사용자 생성의 예를 보이고 있다. 3.1절에서 설명되어진 바와 같이 사용자의 생성은 서버에서 관리자가 직접 설정을 해주어야 한다. 사용자의 생성정보를 관리자가 직접 관리하고 파일 정보들을 관리해주어야 하기 때문에 사전에 프로젝트가 시작되기 전에 정보를 서버에 입력시켜 놓아야 서로의 설계 프로젝트 파일을 공유할 수 있다.

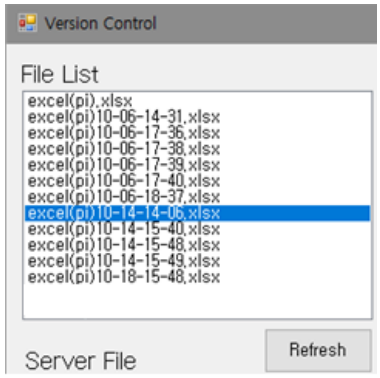


그림 9. 파일 목록 창
Fig. 9. File List Window

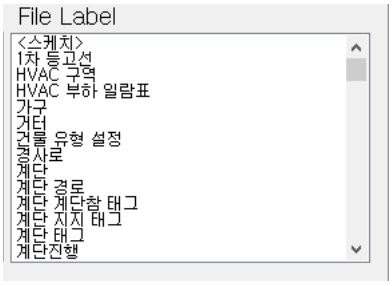


그림 10. 파일 라벨 창
Fig. 10. File Label Window

연결이 완료되면 Refresh버튼을 사용하여 서버에 저장되어 있는 파일 목록을 불러온다. 파일목록은 서버에서 관리자가 부여한 사용자의 아이디에 따라 보여 지는 폴더가 다르기 때문에 각각의 사용자에 따라 파일 목록이 다르고 올릴 수 있는 파일의 이름도 프로젝트에 맞는 파일만 올릴 수 있도록 설정된다. 이때 파일의 이름은 3.2절에서 설명되어진 바와 같이 이름, 사용자명, 월-일-시-분, 의 형태를 띄워 파일 간에 언제 파일이 저장되었는지 누가 이 파일을 수정하였는지 알 수 있도록 표기된다. 또한 이러한 여러 버전의 파일들은 그림에서 보이듯이 자동으로 시간의 순서대로 정렬되는 것을 그림 9를 통해 알 수 있다. 그림 9의 경우 파일이름이 excel이고 pi라는 사용자가 10월 6일 14시 31분 에 파일을 두 번째로 올린 것을 알 수 있다.

이 후 File List에서 파일을 선택하면 그림 10과 같이 File Label 창에 파일의 모든 시트 값들이 출력이 된다. 이때 원하는 시트 값을 클릭하게 되면 원하는 정보를 출력하여 볼 수 있게 된다.

그림 11. 서버 파일 창
Fig. 11. Server File Window

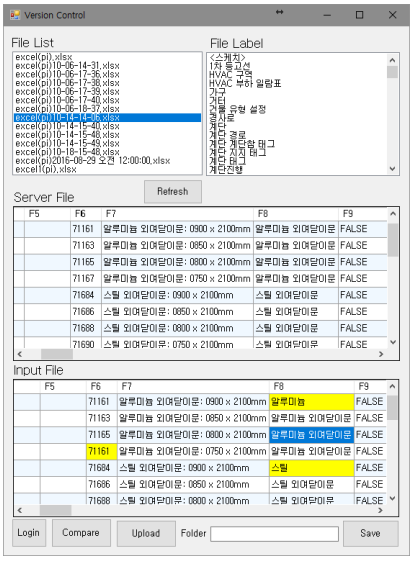


그림 12. 파일 비교 후 프로그램 인터페이스의 모습
Fig. 12. Program Interface after Comparing Files

모든 선택이 완료되어지면 그림 11과 같이 서버에 있는 파일의 정보들이 출력되어진다. 이때 기존의 엑셀 파일과 같은 정보 그대로 출력이 되어 지며 수정하고 싶은 부분의 셀을 클릭하여 수정하고 다시 저장할 수 있도록 Save 버튼을 두어 수정된 파일을 다시 저장 할 수도 있도록 하였다.

또한 새로운 파일을 업데이트하기 위해 Upload버튼을 사용하여 수정된 파일을 선택하면 서버에 파일이 올라가게 되고 간단하게 Input File창에 데이터가 표시된다. 이후 서버에 있는 파일과 비교하기 위해 Compare 버튼을 사용하면 서버에 있는 선택한 파일과 올리려고 하는 파일 간에 비교를 시작하고 다른 부분은 노란색 음영으로 나타나게 된다.

그림 12는 프로그램을 전부 사용하고 비교 부분까지 진행했을 때에 프로그램 화면을 나타내고 있다.

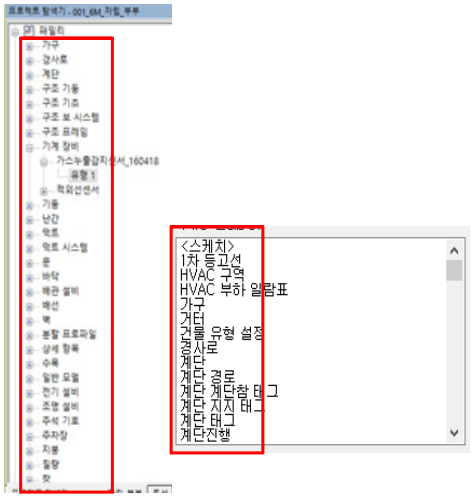


그림 13. Revit의 설계도면 파일과 출력 파일비교
Fig. 13. Comparison of Design Drawing Files and Output Files in Revit

2. 프로그램 파일 검증

프로그램에서 사용되어지는 Excel 파일을 검증하기 위해 실제로 서버에서 다운받은 파일의 정보가 Revit 파일의 설계 도면과 같은지 비교분석하고 정상적으로 저장이 되는지에 대한 비교를 한다. 또한 정상적으로 데이터의 값이 다른 부분을 오류 없이 파악 하는지에 대해서 그림 13과 같이 파악할 수 있다.

Ⅵ. 결 론

선행 연구로부터 Revit BIM 프로그램의 설계 패밀리의 출력물을 엑셀 파일의 형태로 추출가능하게 함으로써 본 논문에서는 이를 기반으로 다수가 같은 설계 도면을 동시에 수행할 때 발생되는 혼동을 효율적으로 관리하기 위한 설계 패밀리 파일들에 대한 버전 관리 시스템을 구현하였다. 또한 그 시스템의 일부분으로 기존의 KMP 알고리즘을 개선한 알고리즘을 제안 구현 하였으며 이를 통하여 좀 더 신속하게 방대한 양의 설계 패밀리 파일들을 비교 검토 하여 수정된 부분이 표시 되도록 하였다.

이러한 시스템을 통하여 기존의 순차적인 설계 프로세서를 병행적으로 효율적으로 진행되어질 수 있을 것이다.

References

[1] Dharmalingam, Vinoth, et al. "System and method for virtual region based access control operations using BIM." U.S. Patent No. 9,342,223. 17 May 2016.

[2] Jong Sung Won, Jung Ju Lee, and Kang Lee. "A Case Study On BIM Collaboration and Information Management Methods." Journal of the Architectural Institute of Korea Planning & Design 24.8, (2008): 25-32.

[3] Dae Ken Guen, Un Jun Sim, and Young Sun Ahn. "A Study on Necessity of Applying of BIM for Maintenance of the Long-Life Housing." Journal of the Korean Institute of Building Construction, 13.2 (2013): 108-109.

[4] Kyoung-Hun Kim, Young-Jae Song. "Component Selection Decision Method Using ANP Technique in Change Management." Journal of the Korea Contents Association, 12.1, (2012): 59-67, 10.5392/JKCA.2012.12.01.059.

[5] Hong-Min Bae, Byung-Seo Kim "Implementation of Excel Export Program for BIM-based Collaborative Design" The Journal of The Institute of Internet, Broadcasting and Communication(JIIBC) 17.1, (2017): 183-190, 10.7236/JIIBC.2017.17.1.183

[6] Eunsang Kim, Jin Wook Kim and Kunsoo Park. "String Matching Algorithm on Multi-byte Character Set Texts." Journal of KIISE : Computing Practices and Letters 16.10, (2010): 1015-1019.

[7] Thomas H. Cormen. "Introduction to Algorithms" MIT Press, 2009. 7. 31.

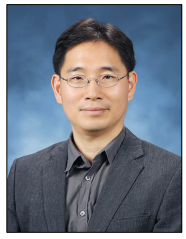
저자 소개

배 홍 민(정회원)

- 2015년 : 홍익대학교 컴퓨터정보통신공학과(공학사)
- 2017년 : 홍익대학교 대학원 전자전산공학과(공학석사)
- 2017년 ~ 현재 : Nikken Sekkei in Japan IT 엔지니어
- E-Mail : hminbae@naver.com

<주관심분야 : IT기술, IoT, WLAN, etc.>

김 병 서(종신회원)



- 1998년 : 인하대학교 전기공학과 공학사
- 2001년 : University of Florida, Dept. Electrical and Computer Engineering M.S.
- 2004년 : University of Florida, Dept. of Electrical and Computer Engineering Ph.D.
- 2007년 ~ 현재 : 홍익대학교 컴퓨터정보통신공학과 교수
- E-Mail : jsnbs@hongik.ac.kr

<주관심분야 : 유무선 네트워크, CDN/CCN, 전술통신, IT융합, etc.>

정 재 희(정회원)

- 1996년 서울대학교 건축학과 학사
- 2006년 University of California, Berkeley, Dept. of Architecture. M.Arch.
- 2010 ~ 현재 홍익대학교 건축공학부 부교수
- E-Mail : jhchung0831@hongik.ac.kr

<주관심분야 : 건축설계, 도시설계, 도시재생, 모듈러건축.>

※ 본 연구는 교육부와 한국연구재단의 지역혁신인력양성사업으로 수행된 연구결과이며 (No. 2014H1C1A1066943), 그리고 중소기업청에서 지원하는 2016년도 산학연협력 기술개발사업(No. C0443024)의 연구수행으로 인한 결과물임.