2024년 춘계학술발표대회 : 대학생부문

로그데이터 기반 BIM 설계 작업 분석 자동화 기술 개발

Development of log data-based BIM design work analysis automation technology

○박 인 건* 박 재 호** 김 이 제*** 진 상 윤****
Park, In-Geon Park, Jae-Ho Kim, Yi-Je Chin, Sang-Yoon

Abstract

In the AEC industry, the role of BIM is expanding. The purpose of this study is to automate the analysis of BIM design productivity using log data. Several preparatory tasks were conducted to utilize log data for analysis, including analyzing the types of log data commands and deriving design commands. Among a total of 16 command types, three commands that were meaningful for design work were derived. Furthermore, An algorithm for formalizing unstructured log data and an analysis automation Revit add-in were developed using Python and C#. The aim is to enhance BIM efficiency during the design phase by providing data-driven knowledge discovery.

키워드 : BIM, 로그데이터, BIM 설계 생산성, 분석 자동화
Keywords : BIM, Log data, BIM design productivity, Analysis automation

1. 서론

1.1 연구 배경 및 목적

최근 정부는 건설산업에서의 혁신을 위해 BIM(Building Information Modeling)을 적극적으로 도입하고자 하며, 이를 위해 BIM을 활용한 다양한 기술 개발 및 제도적 개선에 대한 연구를 추진하고 있다. 하지만 BIM 기반 설계 과정에서 업무 효율성 및 생산성에 관한 정량적인 분석 방법과 결과가 부재하다. 따라서 로그데이터를 활용하여 BIM 설계 작업의 생산성을 정량적으로 분석하고자 한다. 로그데이터는 작업자가 소프트웨어를 실행하고 종료하는 데까지 수행한 모든 행위가 시간에 따라 자동으로 기록되는 텍스트 데이터다. 이를 통해 작업자의 실행 명령어, 오류 사항, 시간 및 날짜 등의 정보를 수집할 수 있다. 하지만 로그데이터는 비정형데이터에 속하기 때문에 활용하기 위해서는 데이터 정제 작업이 필요하다. 따라서 설계 생산성과 관련된 정보를 분석하는 과정은 상당한 시간이 요구된다. 이에 본 연구는 BIM 설계 과정에서 발생하는 로그데이터를 분석하여 작업 시간을 자동으로 계산하는 애드인 개발을 통해 BIM 설계 작업 생산성 분석 및 BIM 활성화를 촉진하고자 한다.

1.2 연구 방법

본 연구에서는 실무에서 가장 널리 사용되는 BIM 저작도구인 Revit의 모든 로그데이터 명령어를 분석하였고, 분석 대상인 설계 작업 명령어를 도출하였다. 그리고 생산성 분석을 위해 Python을 사용하여 비정형인 로그데이터를 정형화된 CSV 형식으로 변환하는 알고리즘을 구현하였다. 이를 바탕으로 C#기반 Revit API를 활용한 생산성 분석 자동화 애드인 프로토타입을 개발하였다. 이때 BIM 로그데이터 관련 연구 고찰을 통해 일부 방법론을 본 연구에 적용하였고, 분석 방법을 구체화하였다. 본 자동화 프로그램이 제공하는 기능을 통해 데이터 기반의 의사결정이 가능해지며, 이에 대한 몇 가지 활용 방안을 제시하였다. 이는 향후 BIM 설계 생산성과 관련된 연구에서 활용될 수 있으며, 분석 결과를 바탕으로 추가적인 생산성 확보 방안을 마련하여 BIM 설계의 효율성을 높일 수 있다.

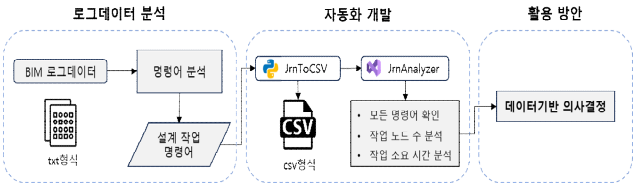

그림1. 연구 플로우차트

* 성균관대 건설환경공학부 학사과정
** 성균관대 글로벌스마트시티융합전공 석사과정
*** 성균관대 미래도시융합공학과 박사 후 연구원, 교신저자
**** 성균관대 건설환경공학부 교수
(Corresponding author : Dept.of convergence Engineering for future city, SungKyunKwan University, yije89@gmail.com)

이 연구는 국토교통부의 스마트시티 혁신인재육성사업과 국토교통부/국토교통과학기술진흥원의 2024년도 지원으로 수행되었음 (과제번호: RS-2021-KA163269).

2. BIM 로그데이터 분석

2.1 Revit 로그데이터 명령어 분석

Revit은 Journal이라는 txt 파일의 로그데이터를 제공한다. 표1은 Revit에서 발생할 수 있는 모든 로그데이터 명령어의 유형, 의미, 예시를 정리한 표이다. 설계 생산성 분석을 위한 로그데이터를 명확히 구분하기 위해 총 16가지의 유형 중 프로젝트의 정보 및 메모리 사용량 등을 기록하는 명령어는 노이즈로 판단하여 제외하였고, 설계 작업과 직접적인 연관이 있는 명령어 유형으로 대상을 좁힐 수 있었다. 각각 애드인 사용에 대한 데이터, 실행 명령에 대한 데이터, 사용자 인터페이스 작업에 대한 데이터를 의미한다.

표1. Revit 로그데이터 명령어

명령어 유형	의미
JournalAddinEvent	애드인에 의해 저장되는 정보들을 포함하고 있는 라인
JournalAPIMessage	이벤트가 Error인지 Successes인지 기록되는 라인
JournalBasicFileInfo	활성화된 프로젝트의 정보를 제공하는 라인
JournalCommand	Revit에서 실행된 모든 Commands를 기록하는 라인 (명령 ID가 존재하는 명령만 기록됨)
JournalComment	JournalLine(JournalAPIMessage, JournalTimeStamp)등 으로 구분되지 않는 모든 comments들을 기록하는 라인
JournalData	Revit 실행동안 사용된 다양한 데이터의 종류를 기록하는 라인 (API 트랜잭션의 성공 여부, 템플릿, 작업 대화상자 및 버튼 눌렀을 때)
JournalDirective	특정 Revit 설정을 기록하는 라인 (Revit 버전, 사용자 이름, 선택 옵션)
JournalGUIResourceUsage	GUI 사용량을 기록하는 라인
JournalKeyboardEvent	Revit에서 누른 특수 키를 기록하는 라인
JournalMemoryMetrics	메모리 사용량을 기록하는 라인
JournalMiscCommand	저널에 포함되지 않는 라인
JournalMouseEvent	Revit에서 사용된 마우스 동작을 기록하는 라인
JournalSystemInformation	하드웨어에 대한 정보를 기록하는 라인
JournalTimeStamp	각 블록의 시작을 기록하는 라인 (여기서 블록은 저널라인을 구분 짓는 소분류의 개념)
JournalUIEvent	Revit 사용자 인터페이스에서의 작업을 기록하는 라인
JournalWorksharingEvent	중앙 모델에 기록되는 동일한 작업 공유 관련 이벤트를 기록하는 라인

연구 고찰 결과, 대다수 BIM 로그데이터 관련 연구에서 JournalCommand 유형의 명령어만 취급한 것을 확인할 수 있었다. 하지만 그림2에서 볼 수 있듯이 시트 정보 변경과 관련된 설정 작업은 JournalUIEvent 유형에 속하는 Jrn.Grid 명령어로 기록된다. 따라서 정보 변경, 속성 설정 등의 작업은 분석에 활용되지 않았다. 본 연구는 설계 작업 생산성에 초점을 두었기에 이러한 세부 작업까지 고려할 필요가 있다. 따라서 인터페이스 작업을 기록하는 유형이 포함된 총 3가지 유형을 설계 작업 명령어 유형으로 구분하였다.

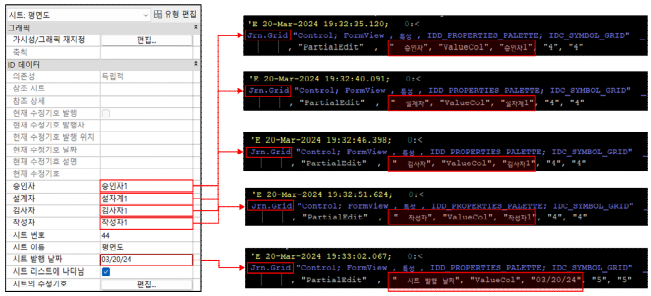

그림2. 설정 작업 로그데이터 예시

Yarmohammadi S et al(2017)에서는 로그데이터 명령어를 상위 명령어와 하위 명령어 개념으로 세분화하였다. 본 연구에서도 원활한 분석을 위해 로그데이터를 명령어의 유형, 상위 명령어와 하위 명령어를 구분하여 분석을 진행하였다.

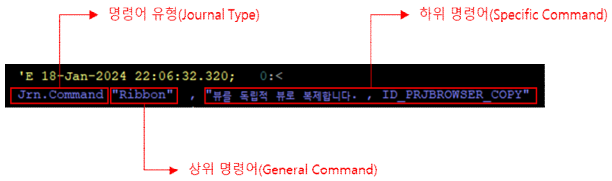

그림3. 로그데이터 명령어 구조

2.2 JrnToCSV: 로그데이터 정형화

Revit에서 제공하는 로그데이터는 대표적인 비정형데이터이기 때문에 불규칙한 공백과 줄 바꿈이 기록된다. 이 데이터를 활용하기 위해선 추가적인 데이터 전처리 작업이 요구된다. 본 연구에서는 2.1에서 도출한 설계 작업 명령어와 그림3의 로그데이터 명령어 구조를 활용하여 비정형 로그데이터 TXT 파일을 CSV 형식으로 정형화시키는 알고리즘을 개발하였다. Python을 사용하여 데이터 정제, 텍스트 분리 및 파싱 작업 등에 대한 규칙 기반 코드를 작성하였으며, 그림5와 같이 설계 작업 명령어에 속하는 로그데이터 CSV 파일이 생성된다. 이를 통해 로그데이터를 직관적으로 이해할 수 있으며, 처음 접하는 사람도 쉽게 분석할 수 있을 것이다. 또한, 3장에서 소개할 프로토타입 모델 내 명령어 확인 기능에 이 알고리즘을 탑재하여 개발하였다.

```python
#데이터 정제하기 Journal_File to Semi_text_file and('[Jrn.'not in line)

def semi_text_file(file_path, target):
    with open(file_path, 'r') as file:
        lines = file.readlines()
    result_lines = []

    for i, line in enumerate(lines):
        if (target in line)and('Jrn.Command "AccelKey"'not in line): #Jrn.Comm
            # 명줄과 해당 줄을 추출
            line_num=str(i)
            command_line = line.strip()
            time_line = lines[i - 1].strip()
            # 다음줄에 ""가 있다면 추출
            while True:
                if '"'in lines[i+1]:
                    next_line = lines[i+1].strip()
                    command_line+=next_line
                    i+=1
                else: break
            result_lines.append((time_line, command_line, line_num))
    return result_lines
```

그림4. JrnToCSV Python 코드 일부

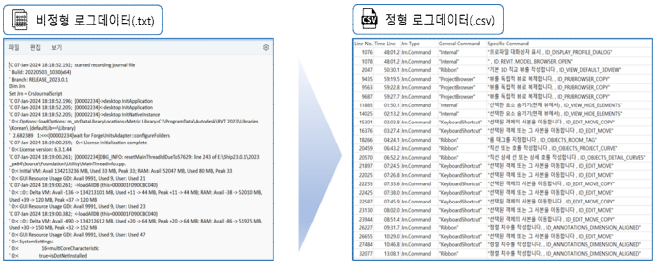

그림5. 로그데이터 정형화 예시

3. 설계 생산성 분석 자동화

3.1 생산성 분석 방법 및 개요

BIM 설계 작업은 모델 데이터를 구축하는 모델링 단계부터 뷰 생성, 주석 작성을 통한 도면화하는 단계까지 모두 포함한다. 이 과정 동안 발생 된 작업 명령어의 개수(작업 노드 수)와 작업이 시작되고 종료될 때까지의 소요 시간을 측정하고자 한다.

작업 노드 수를 파악하기 위해 2.1에서 도출한 3가지 설계 작업 명령어 유형에 속하는 로그데이터 명령어 개수를 카운팅하였다. 이때 작업 명령어의 하위 내용을 분석한 결과, '명령 취소'와 같은 노이즈 명령어와 상태 정보 등을 나타내는 명령어들이 존재하였다. 예를 들어, JournalUIEvent 유형에 속하는 Jrn.Size 명령어는 새로운 창이 활성화될 때 활성 창의 크기 정보를 제공하는 명령어다. 이는 설계 작업을 수행하는 것과 직접적인 관련이 없다고 판단하여 이러한 명령어들을 추가로 제외하였다.

Yarmohammadi S et al(2017)에서는 정확한 시간 측정을 위해 작업 간 공백이 10분 이상 발생한 경우를 연속작업으로 보지 않았다. 따라서 본 연구에서도 작업 간 10분 이상의 공백을 휴식 시간을 간주하였으며, 총작업 소요 시간에서 제외하였다.

3.2 JrnAnalyzer: 생산성 분석 자동화 애드인

2장에서는 BIM 설계 생산성 분석 자동화를 위해 필요한 사전 작업을 수행하였다. 이를 바탕으로 Revit API를 통해 프로토타입 모델을 개발하였다. 애드인은 Revit 2023 버전에서 사용할 수 있으며, 리본 탭에 배치된다. 실행 중인 프로젝트를 대상으로 현재까지의 작업을 분석하기 위한 '현재 작업 파일 열기'와 이전에 작업한 내용을 분석할 수 있는 '파일 직접 찾기'로 파일 열기 버튼을 구성하였다. 후자의 경우 데스크탑 내의 Revit 로그데이터 자동저장 경로로 설정되어 있어 시간 순서로 저장된 파일 중 하나를 선택할 수 있다. 선택된 파일은 아래 '선택된 파일' 메시지 박스에 나타난다. 분석 버튼을 클릭하면 해당하는 핵심 기능이 동작한다. 이때 모든 명령어 확인 기능을 실행하였을 때, 2.2에서 구현한 데이터 전처리 알고리즘이 활용되어 작업 도중 발생한 로그데이터 명령어를 CSV 형식으로 확인할 수 있다. 작업 소요 시간 분석 기능은 로그데이터 내 Timestamp 데이터를 활용하였고, 작업의 종료시각과 시작시각 간의 차이에서 휴식시간을 제외하여 총 소요 시간을 계산한다. 그림7은 애드인의 UI와 결과 화면 예시이다.

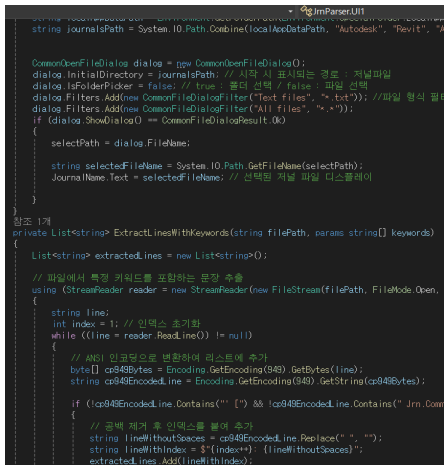

그림6. 애드인 C# 코드 일부

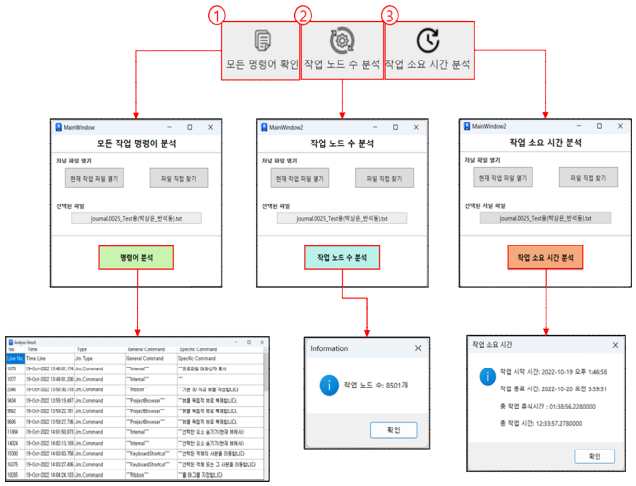

그림7. 애드인 결과 예시

3.3 애드인 활용 방안

로그데이터 기반의 BIM 설계 생산성을 분석은 다음과 같은 방법으로 활용될 수 있다.

첫째, 설계 작업별 효율성 평가

BIM 설계단계에는 객체 모델링, 도면 시트 설정 작업 등 다양한 작업이 존재한다. 이때 동일한 작업자가 단계별 작업에 대한 생산성을 측정함으로써 어떤 작업에서 많은 작업 명령어와 긴 시간이 요구되는지 파악하여 작업 난이도를 구별할 수 있다. 이 결과를 통해 작업별 우선순위 및 사전 투입 계획을 수립하는 등 데이터 기반 의사결정이 가능하다.

둘째, BIM 데이터 상세수준에 따른 생산성 분석

BIM 모델의 상세수준이 높아질수록 작업량도 증가할 것이다. 이에 대한 뒷받침 근거로 로그데이터 기반 정량적인 생산성 분석 결과가 사용될 수 있다. 이때 세부 단계별로 작업 생산성을 분석할 수 있다면, 어느 단계에서 생산성이 저해되는지 파악하고, 생산성 확보 방안까지 마련할 수 있을 것이다.

셋째, 자동화 애드인에 대한 평가

현재 건설산업에서는 BIM 설계 과정을 자동화하기 위한 많은 연구가 수행되고 있다. 대표적으로 도면 생성 자동화, 물량 산출 자동화 등이 있다. 도면 생성 자동화를 예로 들자면, 자동화 애드인의 생산성 분석 결과와 수작업 도면화에 대한 생산성 분석 결과를 비교하여 애드인에 대한 평가가 가능해지고, 효용성을 입증하는 데 활용할 수 있다.

4. 결론

BIM 설계 작업은 복잡하고 창의적인 프로세스로서 대부분 정성적인 분석에 의존한다. 하지만 업무 생산성 및 효율성을 정확히 평가하기 위해서는 정량적인 분석이 필요하다. 이에 본 연구에서는 BIM 로그데이터를 활용하였다. Revit 소프트웨어에서 제공하는 저널 파일을 대상으로 로그데이터 명령어 유형을 분석한 후 설계 작업에 직접적인 영향을 미치는 명령어를 선별하였다. 기존 BIM 로그데이터 기반 연구와 달리 작업 생산성에 초점을 두었기에 세부 설정 작업까지 고려된 작업 명령어 리스트를 도출하였다. 그리고 비정형 텍스트 로그데이터를 분석에 활용하기 위해 Python 기반 데이터 정형화 알고리즘을 개발하였다. 이는 로그데이터에 대한 기본 지식이 없어도 명령어를 분석할 수 있도록 해준다. 3장에서는 이를 바탕으로 개발한 C# 기반 분석 자동화 애드인을 소개하였다. 규칙기반 알고리즘으로 로그데이터 내의 모든 명령어 확인, 작업 노드 수 분석, 작업 소요 시간 분석의 기능을 구현하였다. 현재 개발된 애드인은 프로토타입 모델이며, 향후 개선을 통해 BIM 설계 효율성을 향상하는 도구가 될 것으로 기대된다.

참고문헌

1. 박상은, 로그데이터 기반 BIM 도면 생성 작업의 정량적 분석을 통한 생산성 확보 방안, 석사학위논문, 성균관대 대학원 글로벌스마트시티융합전공, 2022
2. Pan, Y., & Zhang, L. (2020). BIM log mining: Exploring design productivity characteristics. Automation in Construction, 109, 102997
3. Yarmohammadi, S., Pourabolghasem, R., & Castro-Lacouture, D. (2017). Mining implicit 3D modeling patterns from unstructured temporal BIM log text data. Automation in Construction, 81, 17-24
4. Journalysis[github].(2017.11.23.).URL: https://github.com/andydandy74/Journalysis/wiki