creative commons

COMMONS DEED

저작자표시-동일조건변경허락 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.
- 이차적 저작물을 작성할 수 있습니다.
- 이 저작물을 영리 목적으로 이용할 수 있습니다.

다음과 같은 조건을 따라야 합니다:

저작자표시. 귀하는 원저작자를 표시하여야 합니다.

동일조건변경허락. 귀하가 이 저작물을 개작, 변형 또는 가공했을 경우에는, 이 저작물과 동일한 이용허락조건하에서만 배포할 수 있습니다.

- 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

dCollection

공학전문석사학위 연구보고서

건설 공정계획 자동화와 시각화를 위한 프로그램 개발 연구

A Study on Program Development for Automated Construction Scheduling and Visualization

2022년 2월

서울대학교 공학전문대학원
응용공학과 응용공학전공
최흥순

건설 공정계획 자동화와 시각화를 위한 프로그램 개발 연구

A Study on Program Development for Automated Construction Scheduling and Visualization

지도교수 지석호 · 박창우

이 프로젝트 연구보고서를 공학전문석사 학위 연구보고서로 제출함

2022년 2월

서울대학교 공학전문대학원

응용공학과 응용공학전공

최흥순

최흥순의 공학전문석사 학위 연구보고서를 인준함

2022년 2월

위원장 박 문 서 (인)

위 원 지 석 호 (인)

위 원 박 창 우 (인)

초 록

4차 산업혁명 기술과 펜데믹에 의한 뉴노멀 상황은 건설 프로젝트의 불확실성과 변동성, 복잡성을 증대시키며, 건설산업은 다양한 디지털 기술활용과 건설 생산방식의 변화, 프로젝트 관리기법 등에 체질 개선이 요구되고 있다. 본 연구는 변화하는 건설환경에 대응하기 위한 관리기법의 체질개선의 일환으로 차량용 내비게이션과 같이 편리한 건설 공정계획 프로그램 개발을 목적으로 한다. 이를 위해, 기존 공정관리의 문제점과 선행연구를 고찰하고, 기존 연구 성과를 복합적으로 고려하여, 건설공정 내비게이션 개발 방법을 연구한다. 본 연구에서 고안한 가산 그리드 공정계획 기법(CSM)은 공정계획을 자동으로 수립하고 진행현황에 따라 계획을 수정하는 공정계획 기법으로, 평면과 단면의 공간을 일정규격으로 분할하는 그리드를 가지고 작업의 생산성과 후속작업의 투입에 요구되는 선행작업의 완료 영역범위를 계산하는데 활용하고, 작업 위치를 시각화한다. 이 기법을 기반으로 소프트로직 알고리즘 개발과 건설공정 내비게이션의 구성요소와 작동방법을 설명하고, 프로토타입을 개발하여 연구내용을 검증한다. 본 연구로 건설환경 변화와 가변적인 고객 요구에 선제적으로 대응하고 신공법 도입 영향을 빠르게 판단하는 건설 프로젝트 관리 개선에 기여할 수 있기를 기대한다.

주요어 : 건설공정관리, 가산 그리드 공정계획 기법, 소프트로직, 자동화, 시각화, 제약프로그래밍, 그리드, 템플릿, 건설공정 내비게이션

학번 : 2020-25279

목 차

I. 서론 **1**

1.1 연구 배경과 목적 1

1.2 연구 방법과 범위 4

II. 기존의 공정관리 문제점 **6**

2.1 공정관리 전문성의 요구 6

2.2 공정계획 비효율성과 신뢰성 8

2.3 CPM기법의 문제점 10

III. 선행 연구 고찰 **12**

3.1 공정계획 자동화 12

3.2 공간요소 고려 15

3.3 BIM과 공정 연계 17

3.4 소결 18

IV. 건설공정 내비게이션 개발 연구 **19**

4.1 가산 그리드 공정계획 기법 23

4.1.1 CSM의 정의 23

4.1.2 CSM의 특징 25

4.1.3 제안 기법의 효과 28

4.2 소프트로직 알고리즘 개발 31

4.2.1 알고리즘 개발방법 31

4.2.2 알고리즘 개발과정 32

4.2.3 제약조건 정립 42
4.2.4 소프트로직 알고리즘 44
4.3 내비게이션 구성요소 46
4.4 내비게이션 작동방법 49

V. 프로토타입 검증과 투자가치 검토 54
5.1 프로토타입 개발과 구현 검증 55
5.1.1 입력 모듈 55
5.1.2 연산 모듈 58
5.1.3 프로토타입 프로그램 검증과 시각화 60
5.2 프로그램 개발 투자가치 검토 67

VI. 결론 70
6.1 연구 요약 70
6.2 학술적 기여 71
6.3 향후 연구 72

참고 문헌 73

Abstract 77

그 림 목 차

그림 1. Helsman의 복잡도 절벽 2
그림 2. 연구 방법과 범위 5
그림 3. 다양한 수준의 공정표 8
그림 4. 작업현황의 시각화 16
그림 5. 공정관리 프로세스 20
그림 6. CPM과 CSM의 바차트 표현 비교 24
그림 7. 평면 그리드 예시 26
그림 8. 단면 그리드 예시 26
그림 9. CPM 공정표 예시 29
그림 10. CSM 공정표 예시 30
그림 11. 업무별 작업 범위 설정 예시 32
그림 12. 그리드별 작업 순서 배열 33
그림 13. 생산성 제약 적용 35
그림 14. 선행완료거리 제약 적용 37
그림 15. 계약일정 초과 38
그림 16. 완료일정 제약 적용 39
그림 17. 계획대비 실적지연 40
그림 18. 진행 중 - 생산성 조정 요구 40
그림 19. 계획대비 기간단축 41
그림 20. 진행 중 - 생산성 재조정 41
그림 21. 선행완료거리 계산 42
그림 22. 생산성 제약 44

그림 23. 선행작업거리 제약 45
그림 24. 건설공정 내비게이션 구성요소 48
그림 25. 건설공정 내비게이션의 자동화된 프로세스 50
그림 26. 토공사, 기초공사 범위 51
그림 27. 흙막이 업무 범위 51
그림 28. 바차트와 일일작업 시각화 53
그림 29. 입력 모듈 56
그림 30. 그리드별 업무의 상세 작업 분개 57
그림 31. 선후관계템플릿과 업무템플릿 적용 60
그림 32. 그리드별 작업과 순서 배정된 초기 공정표 61
그림 33. 작업순서와 선행완료거리 제약 적용 공정표 61
그림 34. 작업순서와 생산성 제약 적용 공정표 62
그림 35. 작업순서, 생산성 제약, 선행완료거리 제약 적용 공정표 62
그림 36. 연산모듈 결과를 활용한 시각화 적용 공정표 63
그림 37. 초기 계획 64
그림 38. 업데이트 현황과 수정된 공정계획 65

표 목 차

표 1. Automation in construction scheduling: a review of the literature 14
표 2. 차량용 내비게이션과 건설공정 내비게이션 개발방향 . 22
표 3. CPM과 CSM 선후관계비교 27
표 4. CPM 작업기간과 CSM 제약 28
표 5. 알고리즘개발 - 선후관계템플릿 32
표 6. 알고리즘개발 - 작업생산성 34
표 7. 알고리즘개발 - 선행완료거리 36
표 8. 알고리즘개발 - 작업생산성 수정 38
표 9. 건설공정 내비게이션 공정계획 제약 요소 43
표 10. 내비게이션 주요 요소 47
표 11. 연산 모듈 - init 58
표 12. 연산 모듈 update 59
표 13. 절감 비용 및 투자 비용 69

제 1 장

서론

1.1 연구 배경과 목적

최근 4차 산업혁명으로 산업의 생산성의 변화와 펜데믹에 의한 뉴노멀 시대 속에서, 건설산업은 내부적으로 작업 인력의 고령화와 기능인력의 부족 문제가 심화 되고 있고, 로봇 기반의 자동화 기술, OSC(Off-site Construction) 건설기술의 적극적인 도입과 개발이 이루어지고 있다[1]. 그리고 외부적으로는 빅데이터, AI를 이용하는 다양한 서비스와 온라인 시장 확대로 복합화 시설, 물류시설, 데이터 센터와 같은 인프라 구축의 시장 다변화가 일어나고 있다. 이러한 현상은 건설 프로젝트의 불확실성과 변동성, 복잡성을 증대시키며, 건설산업은 다양한 디지털 기술활용과 건설 생산방식의 변화, 프로젝트 관리기법 등에 체질 개선이 요구되고 있다[2].

[그림 1] Helsman의 복잡도 절벽에서 보는 바와 같이 프로젝트 복잡도가 일정 수준이상 높아지면 프로젝트의 성공확률은 현저하게 낮아진다. 복잡도가 높고 다양한 이해관계자가 참여하는 건설 프로젝트를 관리하는데 있어, 공정관리는 프로젝트 정보통합과 이해관계자 간 소통을 위한 기본적인 관리업무 중 하나이다[3]. 그리고 프로젝트 관리자의 합리적인 의사결정을 위해서는 신속·정확한 공정계획 수립이 매우 중요하다[4]. 그러나 현업에 가장 많이 활용하고 있는 CPM(Critical Path Method)과 같은 기존의 공정관리 기법은 전문적인 지식과 전문 도구의 활용 역량을

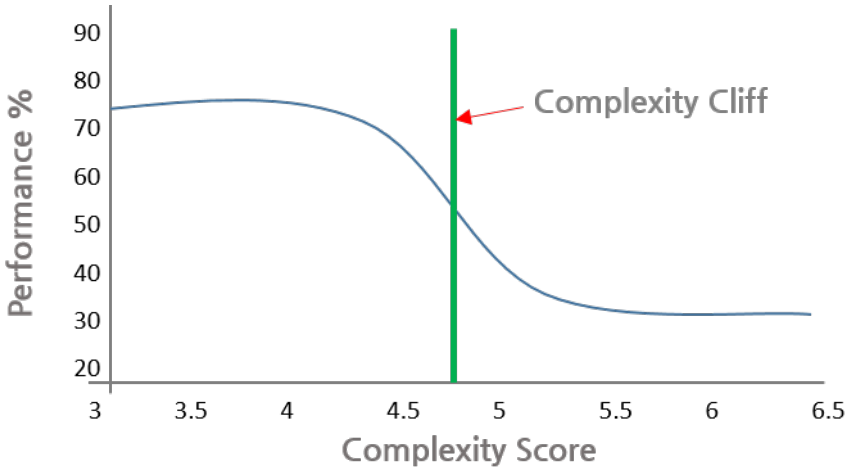

그림 1: Helsman의 복잡도 절벽

필요로 하기 때문에 전문인력과 조직역량을 갖추는데 투자여력에 따라 어려움이 있다[3][5]. 또한, 관리기준 공정표만으로는 작업을 관리하는데 정보가 충분하지 않아 주간 · 월간의 상세 공정계획을 작성하거나 관리자마다 개인 노트나 달력을 이용하여 별도의 기록 활동을 하고 있다[6]. 그리고 공정계획 과정의 많은 부분이 수작업으로 진행되고, 진행에 따른 계획의 변화가 없기 때문에 현장의 변화하는 상황을 반영하는데 많은 한계가 있다[4].

본 연구는 프로젝트 관리기법의 체질 개선의 일환으로, 시장변화에 따른 고객의 요구 변경과 혁신적인 공법 도입에 필요한 빠른 의사결정을 위해, 신속하고 계획의 적정성 판단이 용이한 공정관리 도구 개발을 목적으로 한다. 이를 위해 차량용 내비게이션을 모티브로 하여 비교적 간단한 조작으로 수작업을 최소화하고 공정계획을 자동으로 수립하는 도구인 건설공정 내비게이션을 고안하였다. 이는 전문적인 공정관리 지식이나 공정관리 전문 프로그램 운용 역량을 요하지 않으면서, 건설작업관리에

필요한 일일단위의 상세한 공정계획을 수립한다. 그리고 공사 완료 목표를 달성할 수 있도록 진행현황을 반영하여 잔여계획을 자동으로 수정하여 안내한다. 또한, 공정계획의 적정성을 판단할 수 있도록 작업 내용과 작업 위치를 시각적으로 명확하게 제시한다. 이러한 건설공정 내비게이션을 활용하여 건설환경의 변화를 적극적으로 수용하고, 효율적으로 계획을 수립하고 관리하며, 효과적으로 프로젝트 이해관계자와 소통할 수 있을 것으로 기대한다.

1.2 연구 방법과 범위

본 연구는 건설공정 내비게이션 개발에 필요한 기존과 차별화된 공정관리 기법과 구성요소, 작동방법을 제시하고, 프로토타입 프로그램을 개발하여 제안된 기법과 핵심적인 작동 알고리즘의 활용성을 검증하는 것을 연구 범위로 한다.

이에, 제 2장은 본 연구에서 해결하고자 하는 기존 공정관리 문제점을 선행문헌과 실무경험을 바탕으로 제시한다.

제 3장은 공정관리의 자동화와 공간요소의 고려, 그리고 BIM과 공정연계와 같은 시각화 문제 해결의 선행연구를 고찰한다.

제 4장은 건설공정 내비게이션 개발 연구로, 프로그램 개발에 기초가 되는 '가산 그리드 공정계획 기법(CSM, Countable grid Scheduling Method)을 정의하고 기법의 특징과 효과를 설명한다. 그리고 공정계획을 자동으로 수립하는 알고리즘 개발방법을 설명하고, 건설공정 내비게이션의 구성요소와 작동방법을 제시한다.

제 5장은 엑셀 매크로와 파이썬으로 구현한 프로토타입 프로그램으로 일부 공종의 작업데이터를 반영하여 내비게이션의 구현가능성을 검증한다. 그리고 상용화 프로그램 개발에 요구되는 개략적인 비용을 산출하고, 프로그램으로 얻을 수 있는 기존 대비 절감액을 비교하여 개발 투자가치를 검토한다.

제 6장은 연구성과와 한계점을 논하고 향후 연구방향을 제시한다.

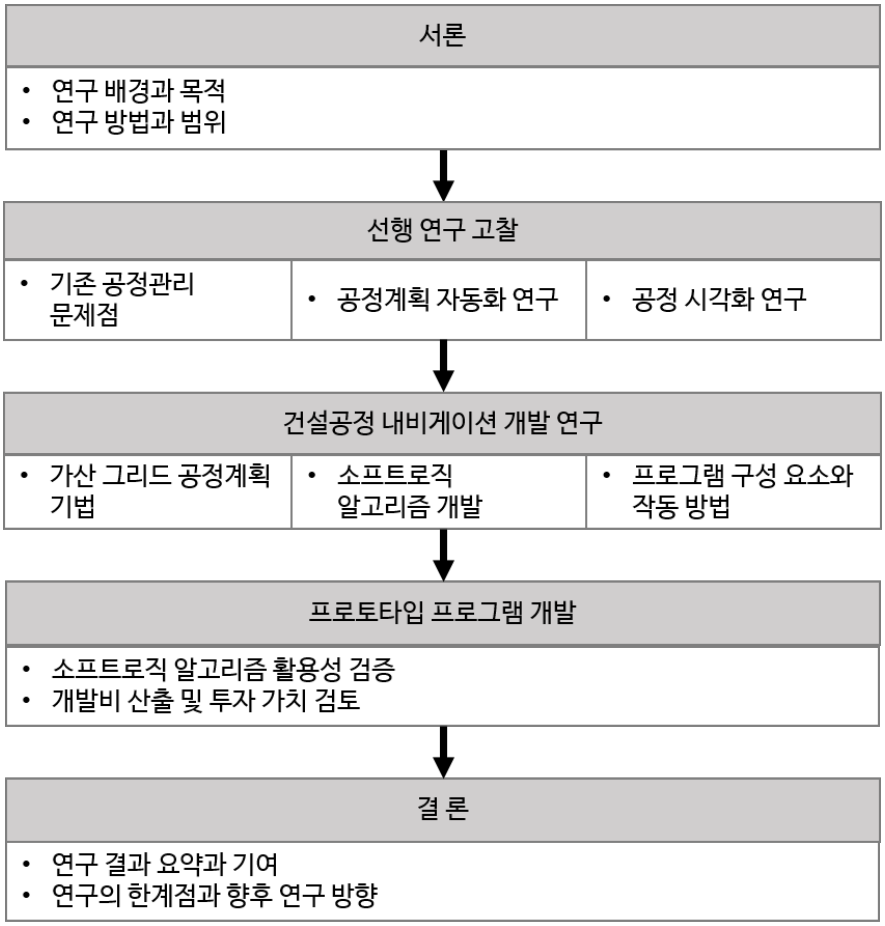
서론
• 연구 배경과 목적
• 연구 방법과 범위
선행 연구 고찰
• 기존 공정관리 문제점
• 공정계획 자동화 연구
• 공정 시각화 연구
건설공정 내비게이션 개발 연구
• 가산 그리드 공정계획 기법
• 소프트로직 알고리즘 개발
• 프로그램 구성 요소와 작동 방법
프로토타입 프로그램 개발
• 소프트로직 알고리즘 활용성 검증
• 개발비 산출 및 투자 가치 검토
결 론
• 연구 결과 요약과 기여
• 연구의 한계점과 향후 연구 방향

그림 2: 연구 방법과 범위

제 2 장

기존의 공정관리 문제점

건설 프로젝트를 성공적으로 관리하기 위해 공정관리는 가장 기본적이고 중요한 관리요소이지만 현업에 공정관리를 운영하는데에 여러 문제점이 있다. 본 장에서는 본 연구에서 해결하고자 하는 기존 공정관리의 문제점을 선행문헌과 본 연구원의 실무 경험을 바탕으로 제시한다.

2.1 공정관리 전문성의 요구

과학적인 공정관리 기법을 적용하기 위해서는 공정관리 지식을 이해하는 전문가와 전문도구의 활용이 요구된다. 장명훈(2016)은 중소규모 건설회사를 대상으로, 허영기(2015)는 부산지역건설사를 대상으로 공정관리 현황을 조사하였다. 중소건설사와 지역건설사는 공통적으로 네트워크 공정표 보다는 바차트를 주로 이용하고 있는데, 그 이유로 전문인력과 조직을 확보하기 위해 인력을 확보하거나 전문조직을 갖추는 것에 부담이 있다고 하였다. 네트워크 공정표는 복잡한 기능을 가진 공정관리 전문 프로그램을 활용해야하고 효율적이지 않기 때문에 엑셀과 같이 간단한 도구로 사용하는 것이 더 효과적이라고 했다. 그리고 상위의 공정표는 네트워크 공정표로 작성했더라도 실제 업무에 활용되는 상세한 공정표는 엑셀과 같은 간단한 도구를 활용하는 것을 선호한다고 했다[5][7].

정영수(2011)는 국내 건설기업이 2000년 초에 비해 CPM이나 LOB (Line of Balance)와 같은 과학적 공정관리 활용도가 높아지고 있고, 공정

관리 업무체계가 고도화되고 전문 인력육성에 많은 투자가 이루어지고 있다고 했다. 그러나 대기업과 중소기업의 공정관리 역량이 차이를 확인할 수 있었고, 사업별로 플랜트에 전문화된 기업이나 해외 사업 비중이 높은 기업이 과학적 공정관리의 활용도가 높다고 하였다. 그리고 실적자료의 재활용과 중소기업의 역량강화는 개선할 부분이 많다고 하였다[3].

김경환(2004)은 프로젝트에 참여하는 이해관계자 간의 의사소통의 중요성과 각 참여자의 전문성의 차이로 공정관리 기법에 대한 이해도에 차이가 있다고 했다. 그래서 실무 활용도를 높이기 위해서는 단순하고 작성이 쉬운 공정관리 방법의 필요성을 강조했다[8].

이처럼 기존의 과학적인 공정관리 기법을 현업에 적용하기 위해서는 전문화된 조직과 전문인력 육성과 같은 투자를 필요로 한다. 이러한 부담은 건설의 효율적인 자원관리와 낭비를 줄이기 위한 과학적 공정관리를 도입하는 데 방해하는 요인으로 볼 수 있다. 따라서 기존의 공정관리 방법보다 이해가 쉽고 사용이 편리한 공정관리 기법과 도구가 필요하다.

2.2 공정계획 비효율성과 신뢰성

건설 프로젝트는 다양한 공종으로 이루어진다. 각 공종은 작업방법, 필요한 자원과 작업 가능한 환경조건이 다양하다. 그래서 각 공종의 작업을 관리하기 위해서는 자재와 작업팀의 준비, 선행작업의 완료일정, 사전 승인 문서와 같은 업무를 누락하지 않고 관리해야 한다. 체계적으로 공정 관리가 운영되는 프로젝트에는 [그림3]와 같이 다양한 수준으로 공정표를 작성하고 관리한다.

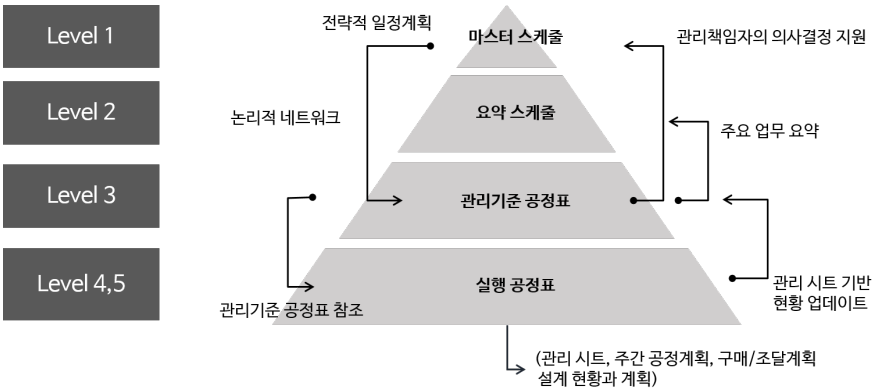

그림 3: 다양한 수준의 공정표

그러나 건설 프로젝트 초기에 작성하는 마스터 스케줄과 관리기준 공정표는 작업을 관리하는데 충분한 정보가 제공되지 못한다. 마스터 스케줄은 프로젝트 전체적인 일정을 한눈에 알아볼 수 있도록, 프로젝트의 주요 일정으로 관리하는 마일스톤의 시점과 전체 공사 일정을 좌우하는 대표 공종의 시작과 종료시점을 볼 수 있는 일정 계획표이다. 관리책임자의 의사결정을 지원하기 위해 문서를 출력했을 때 한페이지에 표현될 수 있도록 작성한다.

관리기준 공정표는 프로젝트 범위를 누락 없이 정의한 업무를 마스터 스케줄의 목표 일정을 참고하여 업무를 수행하는 작업 간에 논리적인 관계를 가지고 공정표로 작성한 것이다. 대부분 전문 공정관리 프로그램을 사용하여 작성하고, 주경로를 기준으로 주요 작업의 목표 일정을 제시한다. 이는 각 작업의 개략적인 시작 · 종료일정과 여유기간, 전체 프로젝트 기간을 좌우하는 주공정 경로를 보여주지만 작업의 정확한 일자를 제공하지 않는다.

그래서 관리기준 공정표를 참고하여 상세한 공정계획을 주간 · 월간 단위로 작성한다. 각 작업이 포함하고 있는 상세한 업무계획을 수립하고, 각 작업의 진행현황을 관리하여 관리기준 공정표를 업데이트 한다. 이때 상세한 계획은 엑셀과 같은 도구를 주로 활용한다.

이현수(2006)는 실무진과 심층면담을 하여 현업의 공정관리 문제점을 조사하고 분석하였다. 현업에 여러 공정표가 운영되고 있고, 각 공종표는 종속적인 관계를 가지고 있다고 했다. 그리고 일정 위주의 공정표는 작업기간의 불확실성과 작업을 관리하는데 필요한 정보가 충분하게 제공하지 못하기 때문에 관리자가 별도로 작업 준비업무와 자재준비 등을 관리할 수 있는 상세공정표가 운영된다고 히였다. 그래서 관리자의 능력 차이에 따른 공정관리 성과차이가 발생하고 있고, 기록관리의 한계로 이미 완료된 공정의 낮은 신뢰성과 공정관리 대상의 불명확성이 기존 공정관리의 효율성과 신뢰성을 저해하는 요인이라고 분석하였다[6].

2.3 CPM기법의 문제점

CPM기법은 건설공사의 기간 준수의 중요성, 일정분석의 용이성의 이유로 사용되어 왔다. 이 기법은 프로젝트 전체기간에 영향을 미치는 주공정을 중점 관리함으로써 전체기간이 늘어나지 않도록 가이드 한다. 그리고 다른 공정관리 기법에 비해 계획과 실적을 비교하고 프로젝트의 공기 지연 원인을 분석하는데 용이하다. 그래서 대부분의 발주처는 공정지연 클레임의 근거를 활용하기 위해 건설사와 계약 조건에 CPM을 적용하도록 하고 있기 때문에 현업에서 가장 많이 활용된다. 그리고 프리마베라나 MS-Project와 같은 전문적인 공정 프로그램은 CPM기법을 적용한다.

하지만 CPM기법은 다양한 측면에서 건설 프로젝트 관리에 적합하지 않다는 지적이 있다. 첫째로, CPM기법으로 건설 공정표를 작성하면 그 형태가 복잡하여 다른 관리자가 이해하기 어려운 문제가 있다. 건설과 같이 프로젝트의 복잡도가 높고 동일한 작업이 반복적으로 진행되거나 다양한 공종이 동시에 진행되는 경우, 복잡한 연결과 대량의 공정표를 작성하게 된다. 그리고 공정계획을 작성하고 수정하는데 많은 시간을 필요로 한다. 그러한 이유로 계획변경이 거의 이루어지지 않는다[9][10].

둘째로, 프로젝트 기간을 좌우하는 주경로 상의 작업을 중점적으로 관리하는 기법의 특성이 문제가 되기도 한다. 복잡도가 높은 프로젝트에서는 주경로가 아닌 일반경로 상의 작업에서 필요한 자재나 기능인력이 확보되지 않거나, 작업 투입 전에 승인받아야 하는 사소한 문서를 누락하거나 필요한 사전 조치 등의 절차를 지키지 않은 등의 다양한 관리 실수로 작업이 진행되지 않아서 일반경로가 주경로가 되어 일정에 영향을 미치기도 한다[6].

셋째로, 공정표를 관리하는 담당자가 공정계획의 의도적인 왜곡하

거나 조작하더라도 이를 즉각적으로 알아차리거나 분석하는 것이 어렵다. CPM기법에서 작업의 선후연결은 선행작업이 완료된 후 후행작업이 시작하는 FS(Finish to Start) 외에 선행작업 시작 후 후행작업이 시작하는 SS(Start to Start), 선행작업 완료 후 후행작업이 완료하는 FF(Finish to Finish), 선행작업 시작 후 후행작업이 완료하는 SF(Start to Finish) 등이 있다. 여기에 Lag와 Lead(-Lag)로 작업 연결 간에 기간을 주는 옵션을 적용할 수 있다. 이는 작업과 작업의 연결 방식을 선택하는데에 개별 작업의 정의에 따라 전문가의 판단이 요구된다[11]. 이러한 이유로, CPM 기법을 활용하여 공정계획을 수립할 때 FS연결을 90%이상 적용하고, 작업기간과 Lag의 기간을 최소화하여 공정계획의 신뢰도를 높일 수 있다고 한다[12]. 하지만 이를 철저히 준수하는 것은 형태가 더 복잡한 공정표를 만들게 된다.

넷째로, 작업을 텍스트로 정의하기 때문에 해석 상에 오해가 발생할 수 있다. 그래서 동사형태로 표현하여 작업내용의 구체적인 표현과 작업 영역을 포함하여 다른 유사작업과 명칭이 중복되지 않도록 하는 것을 권장한다[12]. 그럼에도 시각자료와 같은 보조 도구없이는 프로젝트에 참여하는 이해관계자 간에 소통에 어려움이 있다.

다섯째로, 건설은 작업이 이루어지는 물리적인 공간을 고려한 계획이 필요한데 CPM은 작업 공간요소를 고려지 않는다는 단점이 지적되었다[13][14][15]. 이를 위해 LOB나 바차트와 결합하는 기법 개선과 TACT 공정계획 연구가 있었다[8][16][17]. 하지만 여전히 계획의 작성과 수정이 어렵다는 단점이 존재한다.

제 3 장

선행 연구 고찰

3.1 공정계획 자동화

일반적으로 공정계획의 수립과정은 WBS(Work Breakdown Structure)를 통해 프로젝트 범위를 정의하고, WBS최하위의 업무(Workpackage)와 이를 수행하는 세부작업(Activity)을 추출한다. 각 작업의 물량, 작업 생산성, 투입 자원계획 등의 정보를 바탕으로 작업 기간을 산정하고, 선행작업과 후행작업을 연결하여 공정계획을 작성하는데, 이 과정에서 작업을 선택해서 연결함에 있어 반복과 수정이 많다. 그리고 업무 범위와 작업 구분 기준이 변경되면, 기존의 작업과 기간이 새롭게 정의되어야 하는 불편함이 존재한다.

그래서 선행연구는 반복적인 작업의 추출과 연결에 템플릿을 활용하여 작업을 간소화하였다. 기존에 축적한 실적이나 경험으로 세부적인 설정 값을 저장한 환경을 템플릿이라 한다. 시설이나 용도별로 업무와 세부 작업 그룹을 저장한 템플릿, 작업을 표준화하여 작업 간의 선후관계를 저장한 템플릿을 활용하였다. 이를 이용하면 공정관리 비전문가도 부분적으로 공정계획을 수행할 수 있었다[4].

Zou(2020)은 선후관계가 변경될 수 있는 작업 간 관계를 의미하는 소프트로직에 반복적으로 이루어지는 작업의 연결을 제약조건에 의해 우선순위를 부여하는 알고리즘을 제시하였다. 이를 제약프로그래밍이라고 한다. 제약 조건으로, 1) 작업 위치 간의 이동 시 버퍼시간 부여, 2) 작업

팀의 연속 유지, 3) 작업자의 고용된 시작과 종료기간에 제약을 두었다. 4) 프로젝트의 제한된 기간, 5) 부가 제약을 가지고 개선된 공정계획을 검증하였다. 이를 종합하여 자원의 비용을 최소화하는 최적화 조건에 충족하도록 하였다[18].

공정관리를 자동화하는 연구는 [표 1]과 같이 오랜 기간 이루어져 왔고 다양한 방법이 소개되었다. 기존 실적이나 지식 기반 접근, BIM과 같은 모델 기반 공정추출, 작업간 변이발생으로 개선된 경로를 얻는 유전자 알고리즘, 인간의 신경망을 모사한 뉴럴네트워크, 변수들 간의 관계를 정량적화하여 모델화한 후 시뮬레이션을 통해 최적값을 얻는 시스템다이나믹스 등 다양한 방법으로 공사기간과 원가를 최소화하는 공정 최적화 방안을 제시하였다. 그러나 여전히 수행해야할 작업이 많고 관리자의 공정관리 전문성에 크게 의존할 수 밖에 없다고 하였다. 그래서 스케줄링과 관련된 다양한 방법을 복합적인 방식으로 개선할 필요가 있다[19].

표 1: Automation in construction scheduling: a review of the literature

구 분	내 용	발행 연도
사례 기반 추론 및 지식 기반 접근 방식	유사한 과거 데이터를 활용하고 일정 문제를 해결하기 위한 지식 기반을 개발	1988~2010
모델 기반	공간 추론과 같은 건설 일정을 자동화하기 위한 규칙 개발	1994~2012
유전 알고리즘	CPM과 LOB 통합 컨셉 자원 최적화 및 평준화에 초점	1985~2010
전문가 시스템	AI의 하위 인간의 의사 결정 기술 모방하는 알고리즘, If – then 구조	1970~2001
인공신경망	주로 비용-일정 균형을 강조하고 비용 최적화	1999~2008
기타	System Dynamics Model 등 비용 절감을 극대화 하기 위해 프로젝트 재정렬	2013~2015

3.2 공간요소 고려

건설은 다양한 공종의 작업이 제한된 공간에서 이루어지기 때문에 공간은 공정계획에 중요한 고려 요소이다. 선행작업이 완료되지 않은 상태에서 후속작업이 투입되어 작업 간에 충돌이 발생하거나, 공간에 여유가 있는데도 불구하고 작업이 진행되지 않는 비효율적인 상황이 발생한다[14][20][21]. 그리하여, 공정계획을 시각화하여 작업의 정의를 명확하게 하는 것이 원활한 소통에 필요하다. 그래서 이해관계자와 협의하거나 관리자의 의사결정을 위한 특정 공종의 작업영역을 별도로 인쇄된 도면에 수기로 마킹하거나 [그림4]와 같이 파워포인트 등을 이용하여 작업영역을 표기하고 있다. 하지만 주기적이고 반복적인 업무로 많은 시간이 소요된다.

이와 같은 이유로 작업 공간요소를 고려한 공정관리 방법 연구가 많이 이루어졌다. Thabet(1994)는 공간을 확보해야 하는 자원으로 간주하였다. 공간 특성별로 작업을 분류하고, 작업 요구 공간과 사용 가능한 공간을 비교하여 조건 만족 시 계획을 배정하고 만족하지 않으면 작업 생산성을 높이거나 작업을 분개하는 일정 계획 모델 제시하였다[22].

이현수(2010)는 공간적인 제약 요소에 관한 실태를 분석하고 수직 공간과 수평공간을 고려한 공정계획 모델을 제시했다. 대부분의 건설 현장이의 공간부족 현상보다 공간낭비 현상이 빈번하며, 공간의 중복 작업이 필요함을 지적하고 공간낭비 현상을 줄이는 방안을 제시하였다[15].

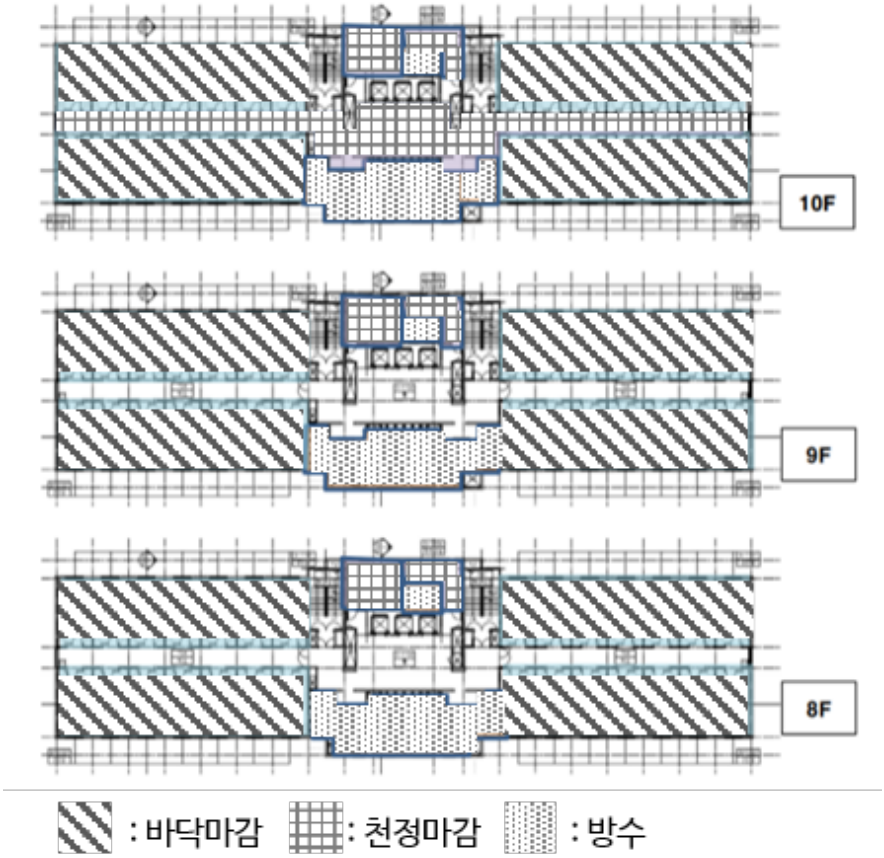

그림 4: 작업현황의 시각화

3.3 BIM과 공정 연계

BIM(Building Information Model)의 활용이 확대되면서 BIM 객체에 다양한 정보를 담아 원가관리와 물량산출, 일정관리를 연계하는 시도가 계속되고 있다. 4D simulation 기술은 BIM 구성요소와 시공방법 정보를 가지고 작업의 순서를 최적화 하기 위한 분석도구로 활용하고 있다. 이는 시각적인 지원으로 여러 이해관계자들의 의견 수렴이나 협력업체의 공종간의 업무를 조율, 공정 비교와 공사 진행 상황을 파악하기가 용이하다[23]. 이에 모델과 공정정보를 수작업으로 연결하는 것을 다양하게 개선하여 효율을 높이고 있다. BIM모델의 객체와 템플릿을 연계하여 모델로부터 작업을 추출하거나 모델 각 객체에 공정정보의 입력을 자동화하는 방법이 있다[24][25][26][27]. 그리고 모델링과 그리드로 시뮬레이션을 효율화하는 방안도 제시되었다[28].

하지만 BIM을 활용하여 공정을 시각화하는데에 다양한 한계점이 존재한다. 첫째로, 건축설계 단계에 시공사가 참여하지 않는 BIM모델링은 시공단계 활용을 고려하기 어렵다. 시공단계에 활용되기 위해서는 작업구역과 골조의 조인트 계획을 고려하여 모델을 분할하여 작성해야 한다. 그래서 설계단계에서 생성된 모델과는 별개로 시공단계에서 BIM모델을 다시 만들거나 모델링으로 부터 추가 정보들을 얻기 위해 시공사가 별도의 객체별 속성정보를 체계적으로 구축해야 한다.

둘째로, BIM모델은 LOD(Level of Detail)100, 200, 300과 같은 상세수준으로 구분하여 작업을 하는데 각 레벨에 따라 표현하는 개체 범위가 다르다. 공정표에 표현하는 작업을 모두 모델로 표현하는 데에는 한계가 있다. 예를 들어 가설설치와 해체, 먹작업, 전기배선 작업와 같은 작업은 대부분 표현하지 않기 때문에 작업은 있으나 모델개체가 존재하지 않을

수 있다[29].

셋째로, BIM을 운영하기 위한 모델링을 작성하고 활용하는데에 필요한 전문 인력을 확보하고, 통합관리를 위한 조직과 거버넌스를 구축이 요구된다. 프로젝트에 참여하는 감리단, 설계사, 시공사가 동일한 프로그램을 사용하도록 하고, 버전관리와 같은 규칙을 준수해야 한다. 그래야만 파일을 공유하고 작성하는데 있어 데이터 호환, 오류발생과 같은 문제발생 가능성을 낮추고 BIM운영에 효과를 높일 수 있다[29].

넷째로, 현재는 BIM으로 설계된 프로젝트가 많지 않고, BIM모델이 없을 경우 공정을 시뮬레이션 하기 위해 2D로 설계된 도면을 BIM모델로 전환이 필요하다. 이러한 모델링을 하는 것과 BIM을 운영하는데 비용적인 부담이 크다는 단점이 존재한다.

3.4 소결

본 연구는 공정계획 자동화를 위한 선행연구 성과 중 템플릿을 활용한 작업 추출방법과 그리드를 활용한 작업영역 구분, 작업의 공간 제약조건, 프로그램밍을 이용한 소프트로직 자동화 방법 등을 복합적으로 고려하였다. 그리고 BIM모델링 없이도 작업의 위치와 내용을 시각적으로 표현할 수 있는 기법을 고안하고 건설공정 내비게이션 개발에 적용하였다.

제 4 장

건설공정 내비게이션 개발 연구

공정관리는 건설 프로젝트를 수행하면서 목표한 예산과 기간을 맞추기 위해 상세한 계획과 현황을 시각화, 수치화하여 프로젝트 관리자의 의사결정을 지원하는 역할을 한다. 그리고 구성원 간 원활한 소통을 지원하여 당면한 문제를 해결하는 프로젝트 관리의 필수요소이다[3].

공정관리는 [그림5]와 같이 계획단계, 공정표 작성단계, 공정 통제단계로 나눌 수 있다.

계획단계는 프로젝트의 범위를 정의, 프로젝트 전체 기간과 주요 목표 일정 수립, WBS의 작성과 작업구역의 구분, 수행계획과 자원계획 수립을 포함한다. 범위의 정의는 MECE(Mutually Exclusive Collectively Exhaustive) 원칙에 준하여 업무를 누락·중복없이 정의하기 위해 WBS를 작성하고 하위의 구성요소가 상위를 충족하도록 한다. 그리고 각 분야의 관리자가 담당 공종별로 수행계획을 수립하여 공정표를 작성하는 데 필요한 논리를 만드는 과정이다.

공정표 작성단계는 세부 업무를 수행하기 위한 작업을 추출하고 각 작업의 자원계획을 고려하여 작업기간을 산정하고, 작업 간 논리적인 연결을 하여 공정표를 작성한다. 그리고 작성된 공정표를 각 분야의 관리자가 검토하고 계획을 확정한다. 이에 프로젝트를 수행하는 건설사가 주도적으로 진행하며, 작성된 공정계획을 감리사 또는 발주처의 승인을 득하는 것을 포함한다.

공정 통제단계는 승인된 공정계획에 따라 진행하면서 현황을 반영하고 계획대비 실시를 비교하여 지연발생에 대비하는 모니터링을 한다. 그리고 범위의 변경이나 수행방법의 변경이 발생했을 때 수정계획을 수립하여 프로젝트의 목표를 달성하기 위한 일련의 과정을 거쳐 종료하는 과정이다.

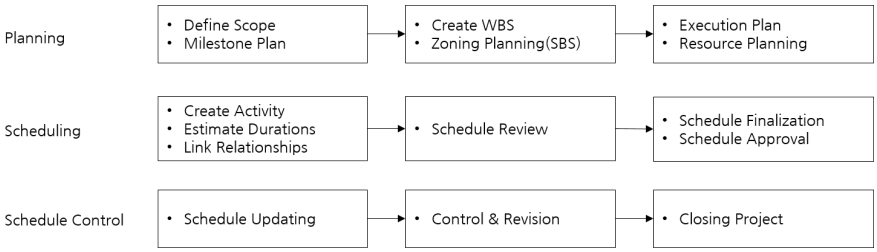

그림 5: 공정관리 프로세스

본 연구는 공정관리 전단계에 필요한 신뢰할 수 있는 공정계획을 신속하게 수립하고, 공정 진행이나 상황 변화에 따라 공정계획을 간편하게 수정하는 공정관리 도구 개발을 목적으로 한다. 이에 차량용 내비게이션으로 부터 아이디어를 발전시켰다. 차량용 내비게이션은 목적지를 입력하는 것만으로, 운전자가 목적지에 도착할 때까지 지속적으로 현황을 반영하고 프로그램된 최적의 경로를 안내한다. 그리고 차량용 내비게이션은 도착 예정시간을 알려주고, 도로의 위험이나 위법에 대비할 수 있도록 경고한다. 차량용 내비게이션의 보급 이전에는, 운전 중에 지도책을 찾아보거나 아는 경로를 이용했지만, 차량용 내비게이션 보급 이후 운전석의 지도책은 사라지고, 내비게이션이 안내하는 경로를 따라 운전하는 것만으로 목적지에 도달하는 편의성을 높였다. 이러한 차량용 내비게이션의 특성을 고려하여 [표2]와 같이 건설공정 내비게이션의 개발 방향을 검토

하였다.

건설공정 내비게이션은 공종별 업무의 맵을 형성하고, 일일단위의 상세한 공정계획을 자동으로 수립하여 그 결과를 시각화된 여러 형태의 공정표로 제시할 수 있는 도구이다. 공사완료 일정에 맞출 수 있도록 지연 위험을 알리고, 작업의 특성에 따른 리스크와 같은 유의할 사항을 안내할 수 있다. 작업의 진행 현황을 반영하여 수정된 계획을 자동으로 갱신한다. 기존에 공정계획을 위해 필요로 하는 전문적인 공정관리 이론이나 프로그램을 활용하지 않아도 공정계획을 수립하고 프로젝트 일정을 준수할 수 있도록 가이드 한다.

본 연구에서는 이를 구현하기 위한 차별화된 공정관리 기법을 제안한다. 그리고 선행 연구의 성과인 작업의 추출과 선후연결 과정을 간소화하는 템플릿의 활용, 그리드를 활용한 작업영역 구분, 작업의 공간 제약, 프로그램밍을 이용한 소프트로직 자동화 방법 등을 복합적으로 고려하였다. 이것들을 활용하여 공정계획 시 수작업을 줄이고, BIM모델링 없이도 작업의 위치와 작업 내용을 시각적으로 표현할 수 있는 프로그램 개발을 연구하였다.

표 2: 차량용 내비게이션과 건설공정 내비게이션 개발방향

구 분	차량 내비게이션	건설공정 내비게이션 개발방향
도구의 목적	목적지로 최적의 경로 안내	수행해야 할 작업을 누락없이 수행가능한 시점에 제시한다.
목표 설정방법	목적지 주소 입력	시각화 도구로 세부공종별 업무범위를 설정한다.
안내 시작	현재의 위치	수행해야 할 모든 작업이 정의된 상태에서 시작한다.
경로 갱신	현재의 위치, 경로변경, 사고발생	잔여 작업을 고려한다. 범위, 공법 변경을 반영한다.
잔여사항	잔여 도로 경로	완료된 작업을 제외한 잔여 작업의 순서를 조정한다.
수행방법	운전(단일작업)	공종별 작업방법(다양한 작업)을 고려한다.
예측 변경 조건	진행 속도, 도로 상황	작업별 생산성, 가설 조건, 업무 범위 변경을 고려한다.
예측	예상 도착시간	예상완료일을 예측하고, 지연시 경고
지연시 단축방법	속도 증가, 우회도로 안내	추가 자원을 투입하거나, 선후관계를 변경한다.
현황 업데이트	실시간 업데이트	일단위로 업데이트 한다.

4.1 가산 그리드 공정계획 기법

4.1.1 CSM의 정의

본 연구에서 제안하는 가산 그리드 공정계획 기법(이하 CSM)은 평면과 단면의 공간을 구분하는 일정한 크기의 그리드를 이용하여 공정계획을 자동화하고 시각화하는 기법이다. 그리드는 공간을 분할하는 역할을 하는 것 뿐 아니라, 그리드의 개수와 그리드 간 거리를 계산하여 공정계획을 수립하기 때문에 가산 그리드(Countable grid)라고 하였다.

이 기법을 활용하면 프로젝트 초기 공정계획을 수립할 때 각 공종의 관리자가 설계도서를 검토하고 각 업무별 범위를 설정하는 것만으로 공정계획을 수립할 수 있다. 이 때 업무별 범위를 설정하는 도구로 일정한 크기의 그리드를 가진 업무별 레이어를 활용한다.

기존 공정관리에 대표적인 기법인 CPM이 주경로(CP, Critical Path)의 지연이 발생하지 않도록 일정 관리를 하는 것에 비해, CSM은 주경로와 일반경로를 구분하지 않고 모든 작업이 적정시점에 진행될 수 있도록 한다. 또한 작업 진행 현황을 반영한 잔여 일정계획을 새롭게 업데이트하여 신뢰도 있는 계획을 유지하도록 한다. CSM은 [그림6]과 같이 CPM과 비교하여 주공정과 일반공정을 구분하지 않고 일일단위의 상세 공정으로 표현한다.

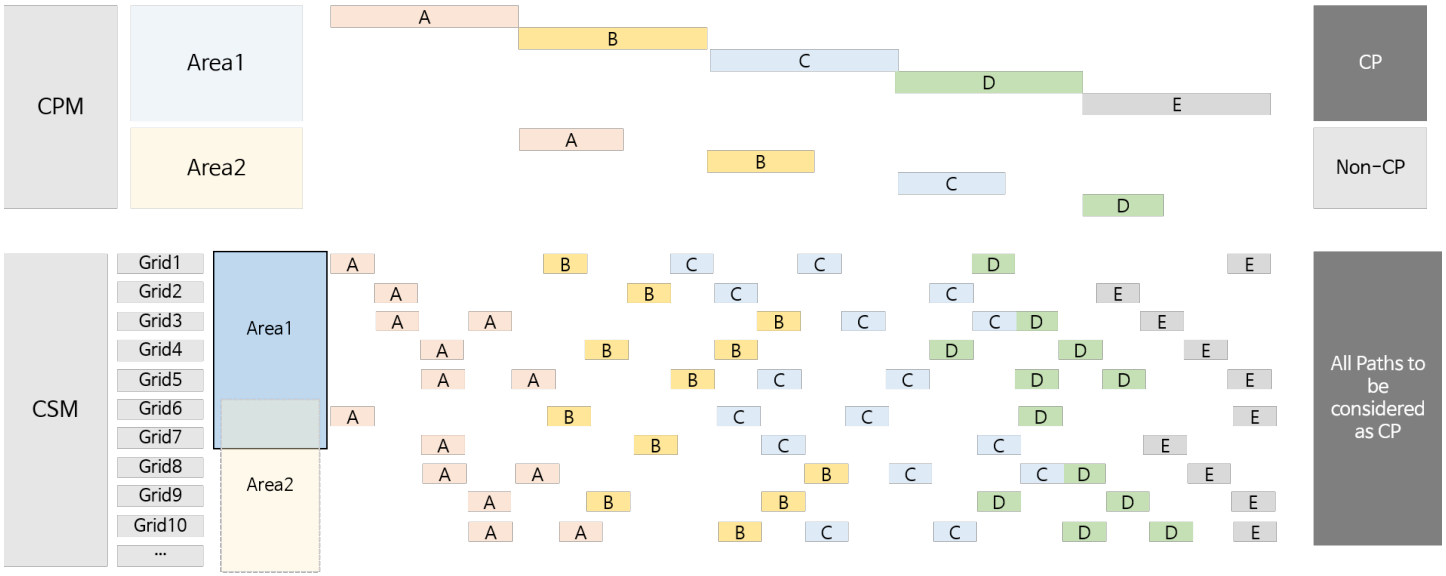

그림 6: CPM과 CSM의 바차트 표현 비교

4.1.2 CSM의 특징

CSM은 그리드를 활용한 시각화된 공간요소 활용, 작업관계의 단순화와 일일단위의 상세한 계획, 계획과 수정의 자동화와 같은 차별화된 특징을 가진다.

첫째로, CSM은 그리드로 작업 영역을 분할하고 작업의 위치를 구체적으로 제시한다. 그리드는 작업영역을 일정 규격으로 세분화하며, 하나의 작업팀이 작업할 수 있는 공간영역을 설명하여 생산성과 선행완료 제약조건의 계산기준이 된다. 그리고 계획된 일정별로 각 업무레이어에 작업 영역을 시각화하는 기준 키(key)값의 역할을 한다. 설계도에서 사용하는 기둥의 중심축으로 작성하는 X축, Y축 그리드와는 연관성은 없다.

[그림 7]과 같이 평면 구역을 사각형의 임의의 동일한 규격으로 구분하고, [그림 8]과 같이 단면 구역은 기본적으로 하나의 층단위로 구획한다. 그리드는 레이어를 활용하여 공종, 구역, 담당자, 시스템 등의 속성을 가질 수 있으며, 위치기반으로 통합한다.

그림 7: 평면 그리드 예시

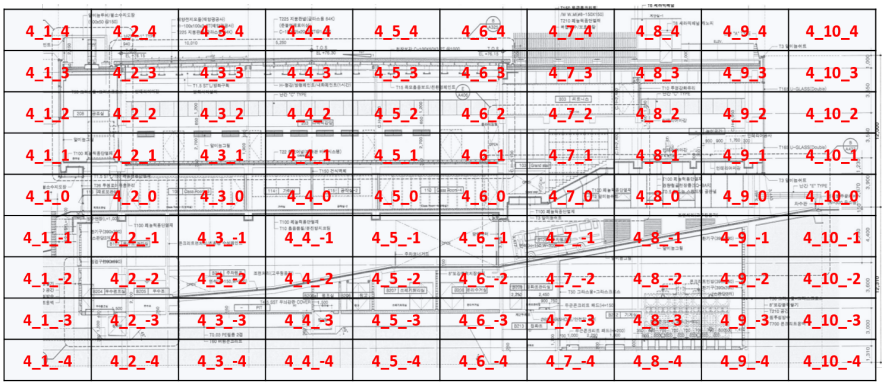

그림 8: 단면 그리드 예시

둘째로, 작업 기간은 1일로 분할하여 정의하며, 작업 간 관계는 FS만으로 연결한다. [표3]과 같이 CSM은 CPM과 달리 SS, FF의 관계나 후행작업의 일정을 지연하거나 앞당기는 Lag나 Lead(-Lag)를 가지지 않는다. 업무를 수행하는 여러 단계의 작업은 하루에 진행가능한 내용으로 분할하여, 모든 작업 기간은 1일로 정의한다. 각 그리드에 배정된 작업 간 관계는 FS이기 때문에 하루에 하나의 작업만 진행가능하다. 그래서 각 그리드에는 작업 간에 공간 충돌이 차단된다. 배정된 작업의 진행 여부를 1(완료)과 0(미완료)으로만 판단하여 그리드 내 완료된 작업을 제거하고 잔여 공정을 연산한다. 이 특징은 CSM이 FS만으로 작업을 연결하여 객관적이고 신뢰도 높은 공정계획을 제시하기 위함이다.

표 3: CPM과 CSM 선후관계비교

Predecessor	Successor	CPM		CSM
		Relationship	Lag	Relationship
Line Marking	Form installation	FS	0	FS
Form installation	Rebar installation	SS,FF	1	FS
Rebar installation	Embedded utility	SS	1	FS
Embedded utility	Inspection	FS	0	FS
Inspection	Concrete casting	FS	0	FS

기존의 CPM에서는 작업과 작업간의 관계가 작업을 분개하는 수준에 따라 달라진다. 소규모 건축물의 경우 층별로 작업을 구분하여 1층 슬래브 타설과 2층 기둥 철근작업이 FS로 연결될 수 있겠지만 1개층의 슬래브 타설이 여러번 나누어 진행될 수 있는 대규모 건축물은 작업 구역을 분할하지 않으면 1층 슬래브 타설과 2층 기둥 철근작업이 SS로 연결될 수 있다. 이를 A구역, B구역과 같이 작업구역을 분할하여 FS관계를 줄 수 있다. 작업을 세세하게 분할하여 FS관계를 가지도록 하면, 진행되는 현황을 반영하여 업데이트 된 잔여작업의 일정 계획과 완료 예측의 신뢰도가 높

아진다. 그래서 가급적 FS관계로 공정표를 작성하는 것을 권장한다[12]. 그리고 SS나 FF연결관계에서 선행작업이 진행하면, 후행작업의 완료 시점을 알기 위해서는 시작한 작업의 잔여기간을 입력해야 한다. 이는 작업관리자의 판단에 의존하기 때문에 의도적으로 일정을 조작하거나 왜곡의 위험이 있다[11]. 그래서 CSM은 신뢰성 있는 일정 예측이 가능하도록 세세하게 공간을 분할하고 FS만으로 작업을 연결한다.

셋째로, 그리드 간 작업순서는 생산성과 선행완료거리 제약조건을 고려한 알고리즘으로 계획하고, 변경된 현황이나 범위의 변경이 업무레이어로 반영되면 자동으로 수정된 계획을 보여준다. 기존의 작업이 작업기간을 가지는 것과 달리 CSM은 [표4]와 같이 하나의 팀이 작업을 하루에 수행가능한 그리드의 개수로 생산성을 정의한다. 그리고 선행완료 거리는 작업이 진행하기 위해 일정거리의 범위에 선행작업이 완료된 공간범위를 의미하여 작업 진행에 필요한 장비나 안전거리 등의 확보를 고려한다.

표 4: CPM 작업기간과 CSM 제약

Activity	CPM	CSM	
	Duration	1-day Productivity	Predecessor completion distance
Line Marking	2d	5 grid	4 grid
Form installation	3d	3 grid	2 grid
Rebar installation	3d	3 grid	2 grid
Embedded utility	1d	9 grid	4 grid
Inspection	1d	9 grid	4 grid
Concrete casting	1d	9 grid	4 grid

4.1.3 제안 기법의 효과

CSM은 세분화된 공간구획의 그리드와 1일단위 작업의 특성으로, 공정계획을 상세하게 계획하고 작업의 위치를 시각적으로 표현할 수 있다.

기존 CPM기법을 적용한 공정표로 보면 [그림 9]와 같이 작업과 일정 그리고 작업간 선후연결관계를 도식화하여 표현한다. 하지만 해당 작업이 어디에서 이루어지는 지 알기 어렵다. 그리고 작업의 이름을 구체적으로 표현하는 것에 한계가 있다. 또한 작업의 기간을 산정하여 입력해야 하고, 선후관계의 연결방법을 작업의 분할 수준과 논리에 따라 FS, SS, FF, SF를 이용하여 정의해야한다.

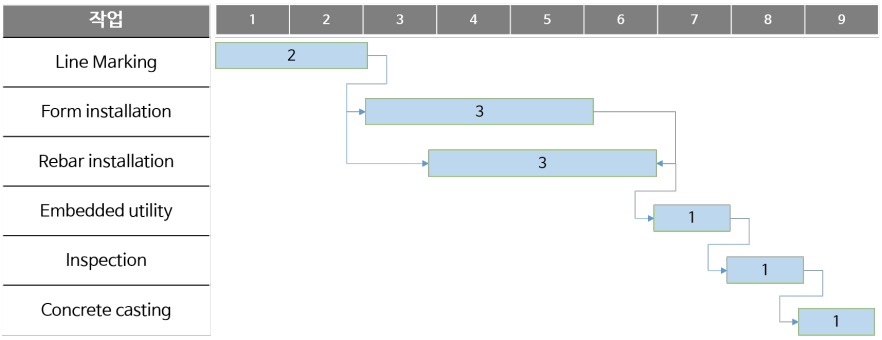

그림 9: CPM 공정표 예시

반면 CSM은 각 작업이 배정된 그리드 범위와 작업간 선후관계, 각 작업이 가지는 생산성과 선행완료거리와 같은 제약 조건으로, 각 일자별로 어느 위치에 어떤 작업이 진행이 될 것인지 명확하게 제시하고 이해관계자 간에 소통할 수 있도록 시각화 한다. 각 그리드에 포함된 작업이 계획된 작업 날짜를 그리드:(작업a:날짜a, 작업b:날짜b, ...) 와 같은 형태로 정보를 생성한다. 그리고 그 정보를 해당 작업을 포함하는 업무레이어에 [그림 10]와 같이 표시한다. 각 작업이 가지는 제약사항을 가지고 소프트로직 알고리즘으로 그리드에 배정된 작업의 우선순위를 배정할 수 있다. 기존의 공정기법은 초기에 수립한 계획을 변경하는 데 어려움이 있었지만, CSM은 공정 현황이나 범위, 방법의 변경에 따라 계획을 자동으로 얻을 수 있다.

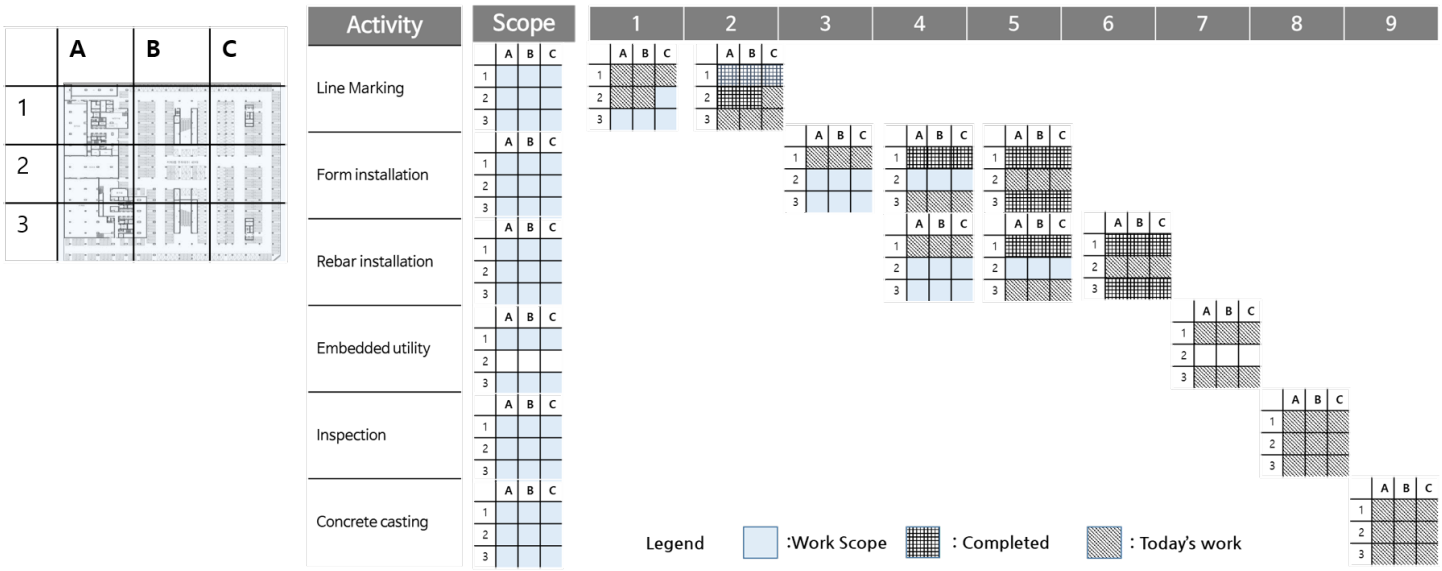

그림 10: CSM 공정표 예시

4.2 소프트로직 알고리즘 개발

4.2.1 알고리즘 개발방법

소프트로직은 작업과 작업 간에 선후관계가 변경될 수 있는 논리를 의미한다. 골조공사에서 1층 이후에 2층 공사가 진행되는 것은 선후관계가 변경될 수 없는 하드로직 관계 또는 필연적 선후관계라고 한다. 이와 달리 천정마감 공사와 바닥마감 공사는 관리자의 계획이나 현장 상황에따라 순서를 달리 진행할 수 있다. 이런 관계를 소프트로직 관계 또는 비필연적 선후관계라고 한다.

공간이 세세하게 분할된 그리드 간에 작업들은 소프트로직 관계에 있고 이를 자동으로 연결하는 알고리즘을 개발한다. 이를 가지고 수행해야 하는 작업이 제약조건에 부합하는 일정에 배정될 수 있도록 한다. CSM을 기반으로, 작업이 하루에 수행가능한 그리드 개수(작업의 생산성), 작업이 투입되기 위해 확보되는 공간(선행작업완료 거리), 그리고 프로젝트가 준수해야 하는 기간을 주요한 제약 조건으로 한다.

4.2.2 알고리즘 개발과정

수동으로 고려할 수 있는 단순한 4개의 그리드로 구획한 평면과 마감공사 작업일부를 가지고 CSM을 기반으로 공정계획을 자동으로 수립하는 과정을 시뮬레이션 하였다.

1) 평면을 그리드로 분할하고 [그림11]과 같이 업무 범위를 배정한다.

각 그리드에 배정된 작업은 [표5]와 같이 선후관계템플릿에서 정의된 선후관계에 따라 작업순서가 정해진다 [그림 12].

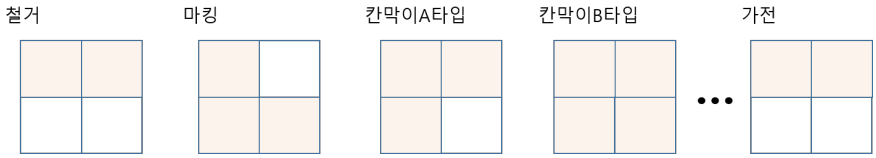

그림 11: 업무별 작업 범위 설정 예시

표 5: 알고리즘개발 - 선후관계템플릿

선행	후행
철거	마킹
마킹	칸막이A
칸막이A	칸막이B
칸막이B	방통
방통	천정
천정	바닥
바닥	도어
도어	가구
가구	가전

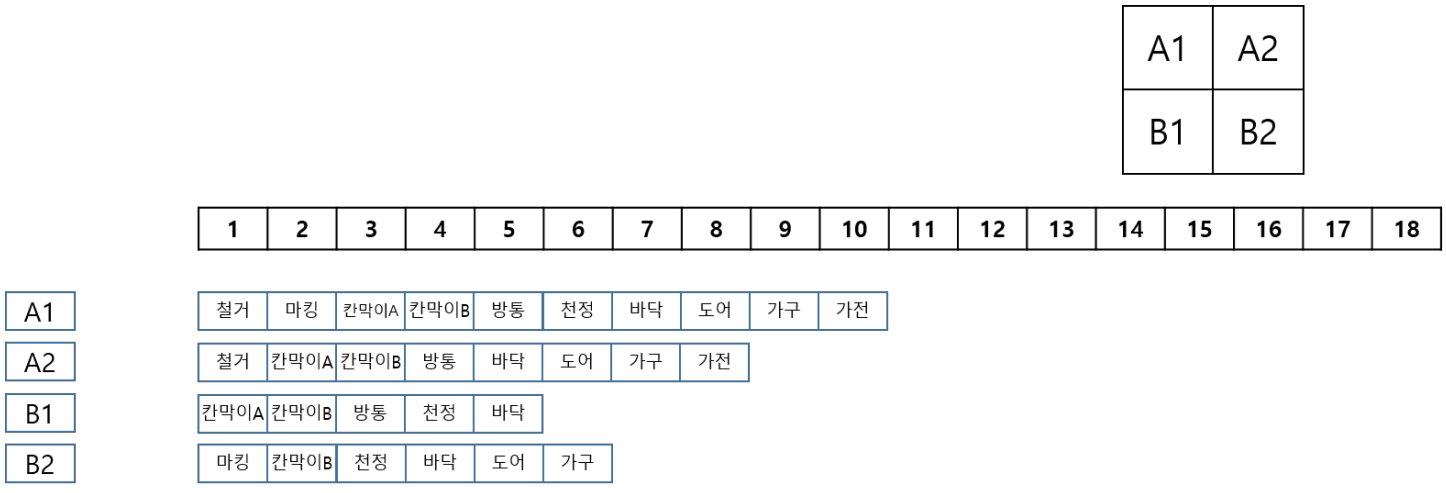

그림 12: 그리드별 작업 순서 배열

2) 그리드별로 배정된 특정의 동일한 작업은 제약된 자원으로 동일한 일정에 수행할 수 있는 생산성 만큼만 진행 가능하다.

[표6]와 같이 하루에 동시에 수행할 수 있는 생산성으로, 철거는 하루 1개 그리드, 마킹은 2개 그리드, 칸막이A는 1개 그리드, 칸막이B는 2개 그리드, 방통 3개 그리드, 천정 1개 그리드, 바닥 2개 그리드, 도어 1개 그리드, 가구 2개 그리드, 가전 1개 그리드로 주어졌다. 생산성제약으로 하루에 진행할 수 있는 작업의 그리드 개수를 제한한다.

따라서 [그림13]과 같이 작업별 생산성에 따라 일정이 조정된다. 철거 작업은 하루에 한개의 그리드 작업만 가능하기때문에 A2그리드의 철거는 하루 밀린다. 칸막이A 작업도 하루에 한개의 그리드만 가능하기 때문에 A2그리드와 B1그리드의 칸막이A 작업이 하루씩 더 미뤄진다. 칸막이B 작업은 하루에 2개의 그리드 작업이 가능하기 때문에 B1그리드, B2그리드에서 동시에 작업이 가능하다.

표 6: 알고리즘개발 - 작업생산성

작업	생산성
철거	1
마킹	2
칸막이A	1
칸막이B	2
방통	3
천정	1
바닥	2
도어	1
가구	2
가전	1

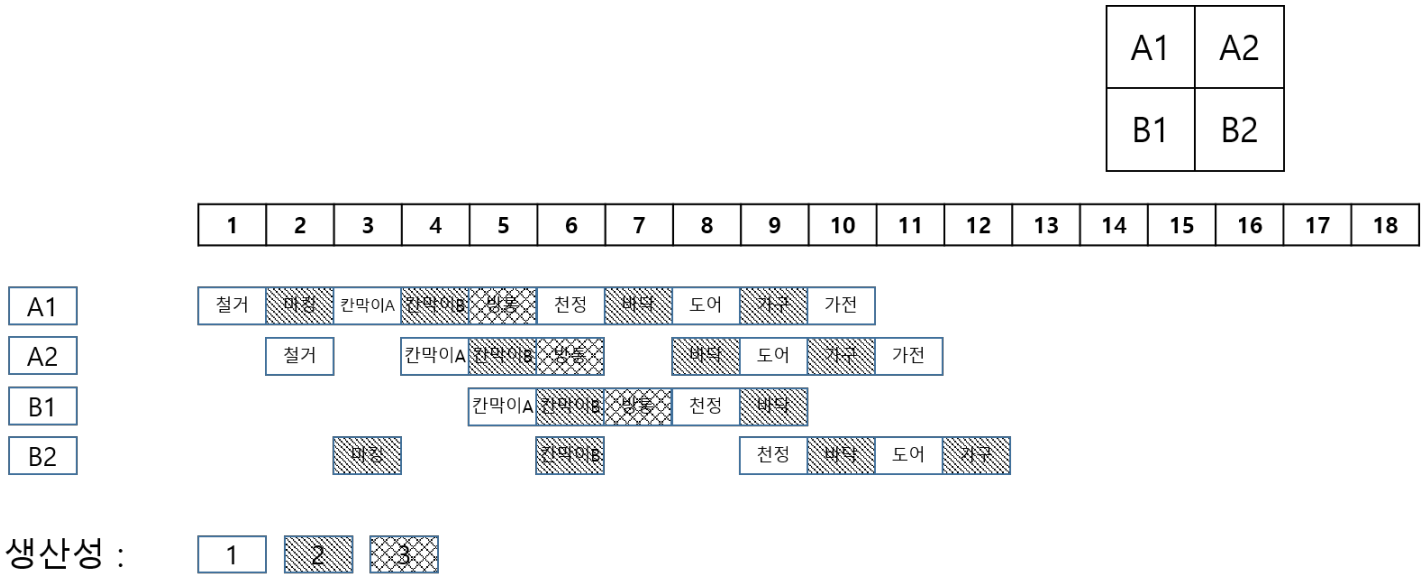

그림 13: 생산성 제약 적용

3) 그리고 작업이 가지는 특성에 따라 자재야적공간이나 작업자 동선, 장비나 작업자의 활동 반경 등을 고려하여 작업이 진행 될 수 있는 공간을 확보해야 한다. 그래서[표7]와 같이 작업이 시작하기 위해 해당 작업 영역 주변 그리드의 선행 작업의 영향을 미치도록 한다. 작업이 진행되기 위해서는 해당 거리 이내에 선행되는 작업이 완료된 상태에서 작업을 투입하게 함으로써 작업 공간과 작업의 최소 물량을 확보하는 효과를 부여할 수 있다.

[그림14]과 같이 작업별로 요구되는 공간을 확보하기 위해서 해당 거리 영향으로 일정이 조정된다. 본 예시에서는 거리1은 변이 맞닿아 있는 그리드, 거리2는 대각선의 위치에 있는 그리드로 정의한다. A1에 마킹장업은 거리 1에 A2의 철거작업이 완료된 후 작업이 가능하여 2일에서 3일로 일정이 조정된다. 방통의 작업은 거리 2이내에 선행되는 칸막이B의 작업이 모두 완료된 이후로 일정이 조정된다.

표 7: 알고리즘개발 - 선행완료거리

작업	선행완료거리
철거	0
마킹	1
칸막이A	0
칸막이B	2
방통	2
천정	0
바닥	1
도어	2
가구	0
가전	0

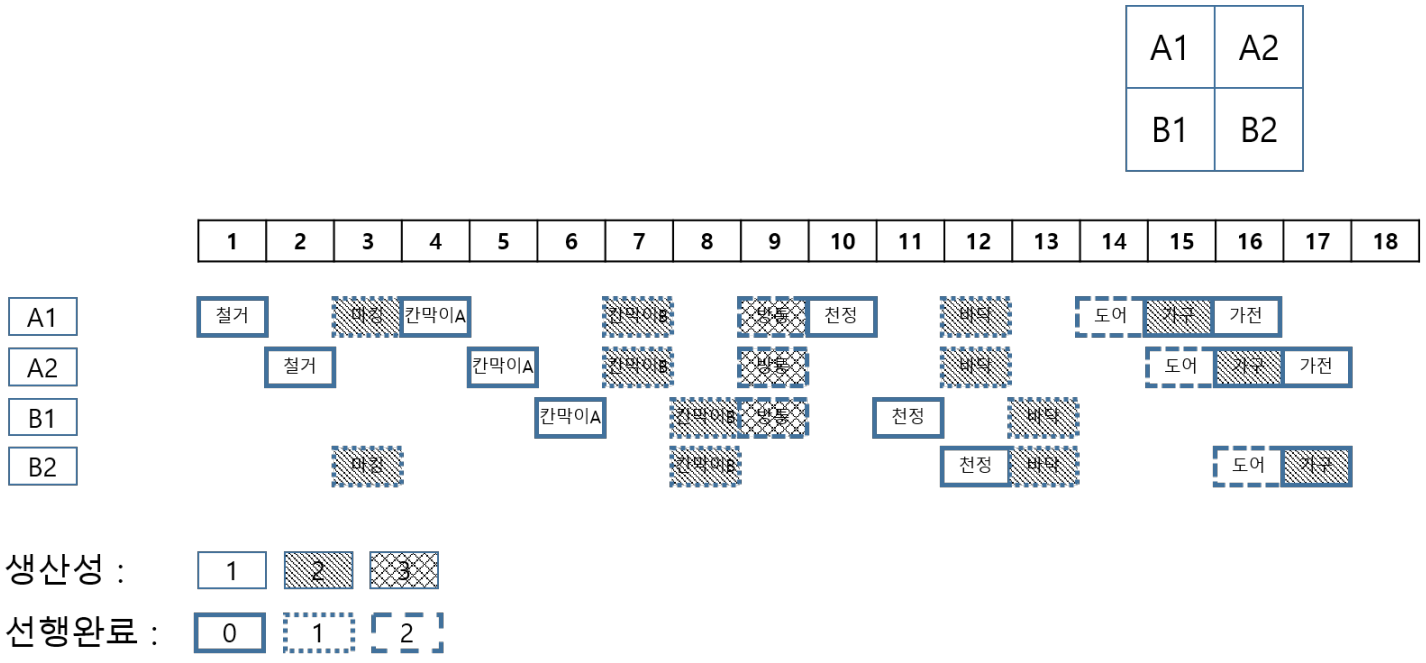

그림 14: 선행완료거리 제약 적용

4) 생산성과 선행완료거리 제약에 의해 조정된 전체 계획일정이 [그림15]과 같이 프로젝트의 계약기간을 초과하는 경우가 발생하면, 프로그램은 알람을 주도록 하고, 관리자가 작업 투입 계획을 수정할 수 있도록 한다.

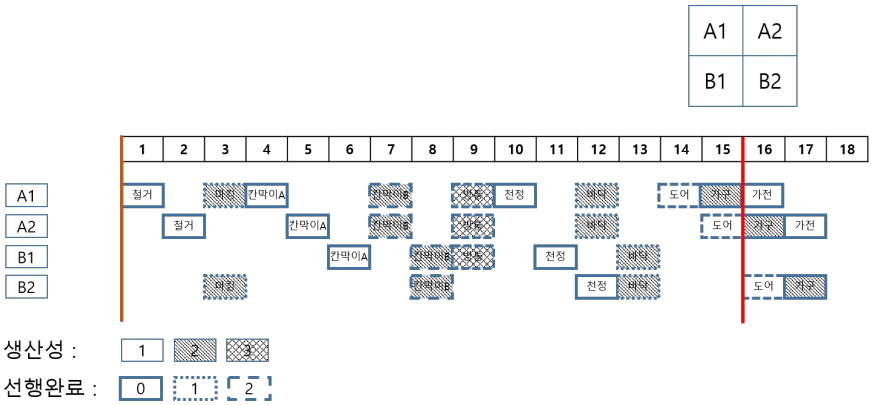

그림 15: 계약일정 초과

그래서 [표8]와 같이 작업 생산성을 조정하거나 투입팀을 늘려서 기간 내에 완료될 수 있도록 한다.

표 8: 알고리즘개발 - 작업생산성 수정

작업	생산성
철거	2(변경)
마킹	2
칸막이A	2(변경)
칸막이B	2
방통	3
천정	1
바닥	2
도어	1
가구	2
가전	1

그 결과 [그림16]과 같이 프로젝트 일정을 준수 할 수 있도록 가이드 한다.

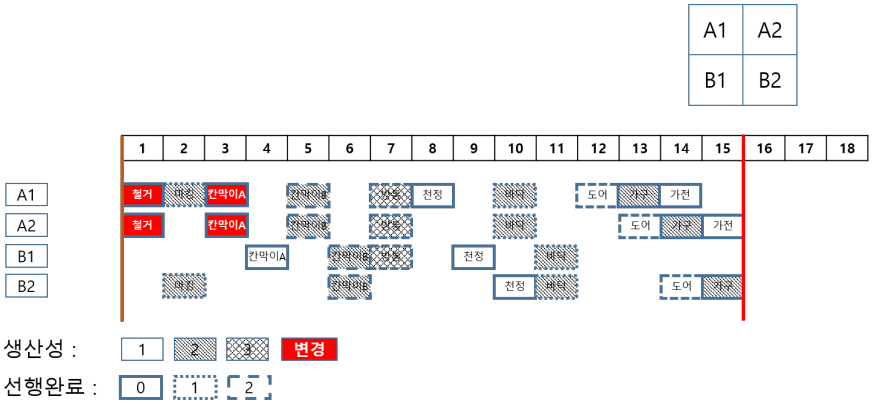

그림 16: 완료일정 제약 적용

5) 초기 공정계획을 수립하고 작업의 진행에 따라 잔여 일정은 소프트로직 알고리즘에 따라 일단위의 실시간 변경된 공정계획을 얻을 수 있다. [그림17]과 같이 작업 진행이 지연되어 전체기간이 밀리게 되면 [그림18]과 같이 잔여작업의 생산성을 조정하도록 한다.

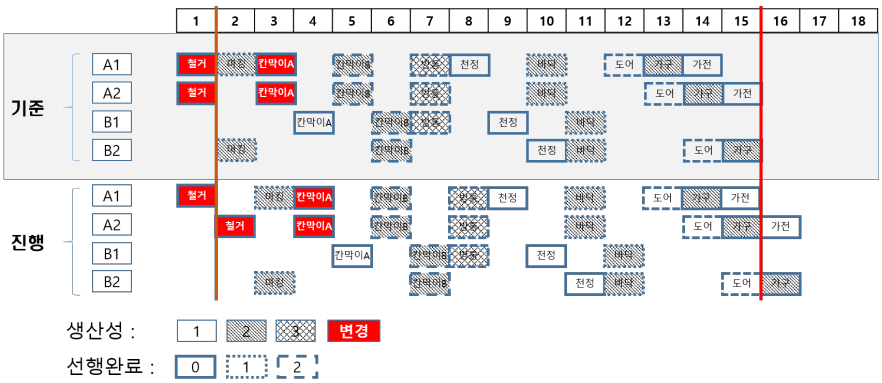

그림 17: 계획대비 실적지연

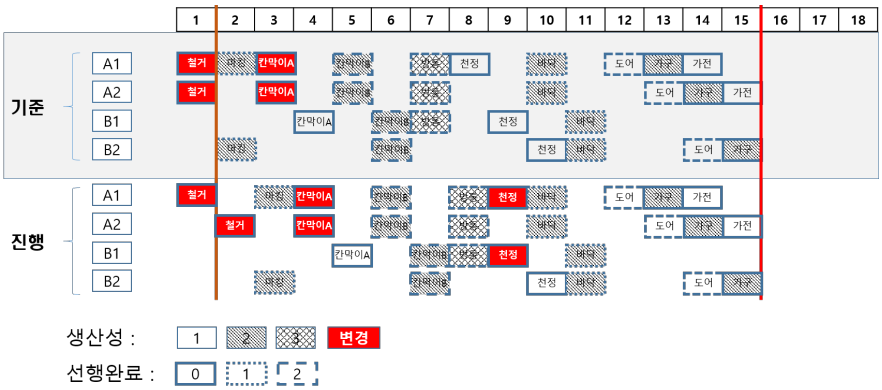

그림 18: 진행 중 - 생산성 조정 요구

6) 반대로 작업이 진행되면서 [그림19]과 같이 전체기간을 앞당기게 되면 [그림20]과 같이 [그림18]에서 추가한 생산성을 제거하여 전체기간을 준수하도록 한다.

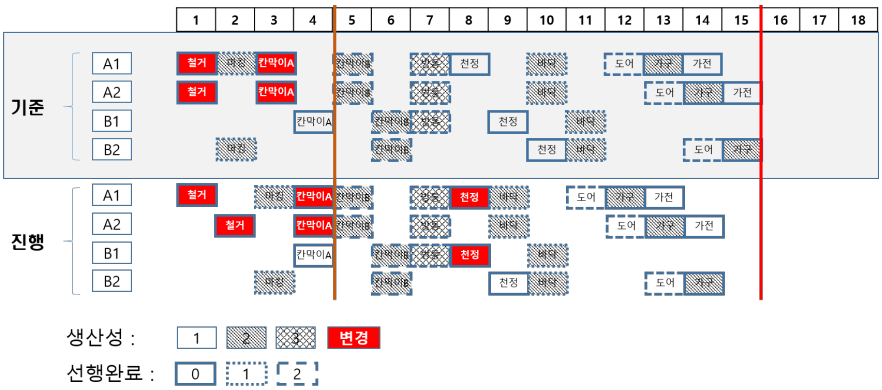

그림 19: 계획대비 기간단축

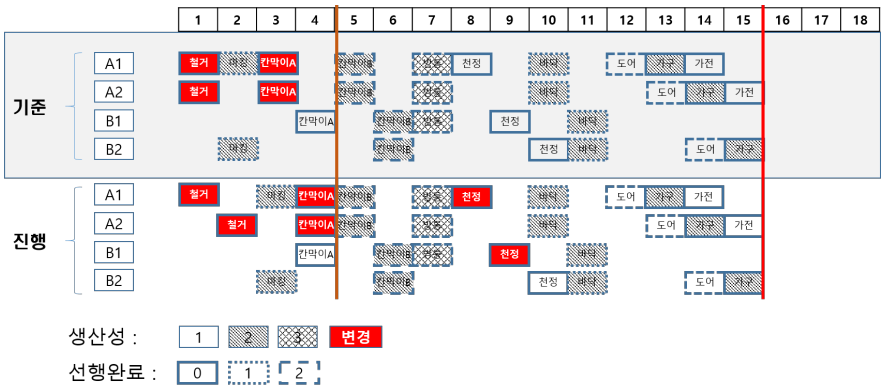

그림 20: 진행 중 - 생산성 재조정

4.2.3 제약조건 정립

위의 내용을 정리하여 건설공정 내비게이션에 적용하는 소프트로직 알고리즘은 [표9]과 같이 총 5개의 제약조건으로 이루어진다.

1) 한 그리드에는 하루에 1개의 작업만 가능하다. 2) 선행작업은 선후행 템플릿에 정의한 순서를 준수한다. 3) 작업이 1개팀에 의해 하루에 진행될 수 있는 그리드의 개수가 작업 생산성이다. 생산성은 투입팀을 조정하면 투입팀 수만큼 배로 증가한다. 4) 작업에 따라 선행작업이 일정거리 이내에서 진행 중이면 해당 작업이 불가하다. 한 팀이 작업할 때 주변 작업과 마찰이 발생하지 않도록 일정거리를 두도록 한다. 작업자와 작업자 간의 거리와 장비와 장비의 안전거리를 확보한다. 거리의 계산은 [그림 21]과 같이 각 축의 거리값 이내의 그리드로 기준한다. 0에 해당하는 그리드에 선행완료거리가 1인 작업은 1로 표시된 그리드에 선행되는 작업이 완료되어 있어야 진행이 가능하여 작업의 날짜가 배정된다. 1로 표시된 그리드에 선행의 작업이 배정되어 있는 경우, 0에 해당하는 그리드에 있는 작업은 하루씩 밀도록 한다.

3	2	2	2	2	2	3
3	2	1	1	1	2	3
3	2	1	0	1	2	3
3	2	1	1	1	2	3
3	2	2	2	2	2	3

그림 21: 선행완료거리 계산

5) 모든 작업은 제한된 기간 안에 완료되어야 한다. 제한된 기간 내 작업이 불가한 경우 자원 계획 변경을 요구한다.

표 9: 건설공정 내비게이션 공정계획 제약 요소

No	제 약	내 용
1	1-1-1 제약	· 하루에 하나의 그리드에는 하나의 작업만 진행한다.
2	작업순서 제약	· 선후관계템플릿에서 정의한 선후 관계를 준수한다.
3	생산성 제약	· 작업별로 하나의 팀이 하루에 할 수 있는 그리드 개수가 제한된다. · 업무 템플릿에 정의한다.
4	선행완료거리 제약	· 작업이 진행되는 그리드와 일정거리 이내에 있는 그리드에 선행되는 작업이 완료되어 있어야 한다. · 업무 템플릿에 정의한다.
5	프로젝트기간 제약	· 모든 작업은 프로젝트 기간 내에 모두 완료되어야 한다. · 프로젝트 기간준수가 불가한 경우, 작업팀수의 조정과 같은 생산성 조정 알림을 준다.

4.2.4 소프트로직 알고리즘

5가지의 제약조건 중 계획의 자동화를 위해서 컴퓨터가 반복적으로 연산을 해야 하는 제약은 생산성 제약과 선행완료 거리 제약이다. 이를 프로그램으로 구현하기 위해서 수동으로 각 작업의 일정을 조정하는 단계를 검토하고 파이썬으로 구현하면서 [그림 22]와 같이 생산성 제약 연산 알고리즘과 [그림23]과 같이 선행완료거리 제약 연산 알고리즘을 개발하였다.

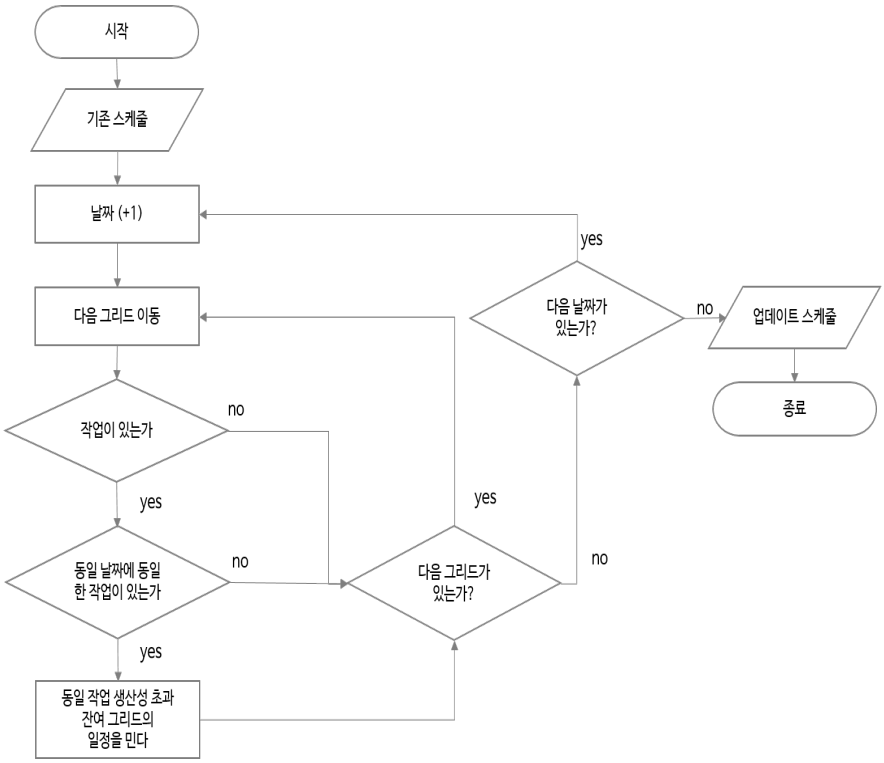

그림 22: 생산성 제약

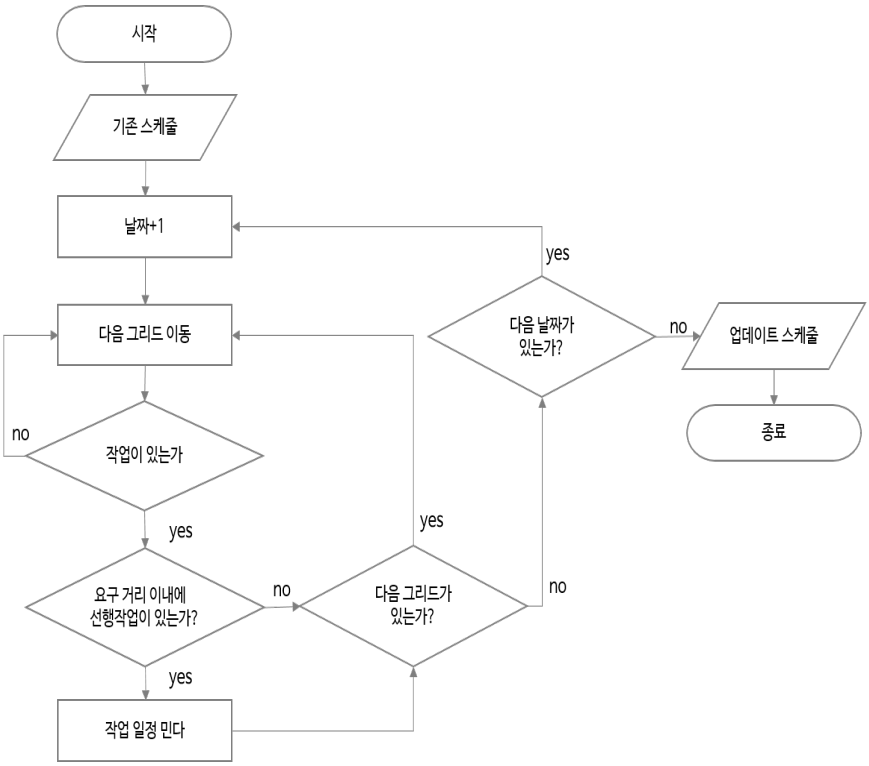

그림 23: 선행작업거리 제약

4.3 내비게이션 구성요소

건설공정 내비게이션은 [표10]과 [그림24]와 같이 입력모듈, DB모듈, 연산모듈, 출력모듈로 구성된다. 입력모듈은 2D도면과 업무레이어로 구성된다. 업무레이어는 업무범위를 시각화하고, 업무범위와 업무템플릿을 연결하는 도구로써, 업무레이어에 업무(Workpackage)별 범위를 그리드 단위로 입력한다. 입력된 영역에 각 업무는 업무템플릿에 정의된 작업이 배정된다. 이 때 업무템플릿의 작업그룹으로 작업배정을 간소화한다. 일반업무레이어, 가설레이어, 실별레이어 등으로 업무템플릿 구성별로 분류하여 활용할 수 있다. 업무레이어는 토공사, 골조공사, 칸막이벽공사, 외장공사 등의 공종으로 구성된다. 가설레이어는 가시설, 이동통로, 양중장비 등으로 구성할 수 있다. 실레이어는 마감공사 업무레이어를 실별 마감내용을 달리하여 업무템플릿을 구성하여 작업 범위 설정의 편의성을 높일 수 있다.

DB모듈은 업무템플릿과 선후관계템플릿으로 구성된다. 업무템플릿은 업무별로 세부 작업을 정의하고 작업별 생산성과 선행완료거리 조건과 같은 작업의 속성을 정의하는 도구이다. 위의 속성 외에도 자원과 리스크, 관련문서 등 작업에 필요한 정보를 확장하여 다양한 데이터 분석과 업무자동화를 수행할 수 있는 데이터를 관리 할 수 있다. 그리고 실적데이터를 축적하는 도구로 활용될 수 있다.

선후관계템플릿은 작업 간의 선후관계를 정의한다. 선후템플릿은 선후가 바뀔수 없는 하드로직이나 관리자가 프로젝트에 선후를 지정하고자 하는 작업 간 선후관계를 정의한 테이블형식의 템플릿이다. 기존의 프리마베라에서 추출할 수 있는 관계테이블(Relationship)과 유사하다. CSM은 FS만을 사용하고 Lag가 없기 때문에 선행작업과 후행작업 컬럼만 가

진다.

연산모듈은 그리드 간 작업 우선순위를 자동으로 배정하는 소프트로직 알고리즘이다. 생산성과 선행완료거리 제약조건에 의해 각 그리드별 작업의 일정배정을 연산하고, 프로젝트 기간이나 작업팀의 투입을 조정하는 전략적 제약으로 구성된다.

출력모듈은 연산된 그리드별 작업의 일정 데이터를 바차트로 변환하여 보여준다. 그리고 각 일자별 작업 진행 범위를 업무레이어의 해당그리드에 표시한다.

표 10: 내비게이션 주요 요소

모 듈	주요 요소	내 용
입력모듈	업무레이어	· 그리드를 가진 업무별 범위 설정 도구 · 일반업무, 가설, 실, 공구 등 다양한 구분에 활용
DB모듈	업무템플릿 선후관계템플릿	· 업무별 세부작업과 각 작업의 생산성, 선행완료거리 등 속성 정의 · 작업 간 하드로직관계 정의
연산모듈	소프트로직 알고리즘	· 1-1-1 제약 · 선후관계 제약 · 생산성 제약 · 선행완료거리 제약 · 프로젝트 기간 제약 연산
출력모듈	업무레이어 바차트	· 그리드별 일자별 작업을 바차트와 업무레이어에 표시

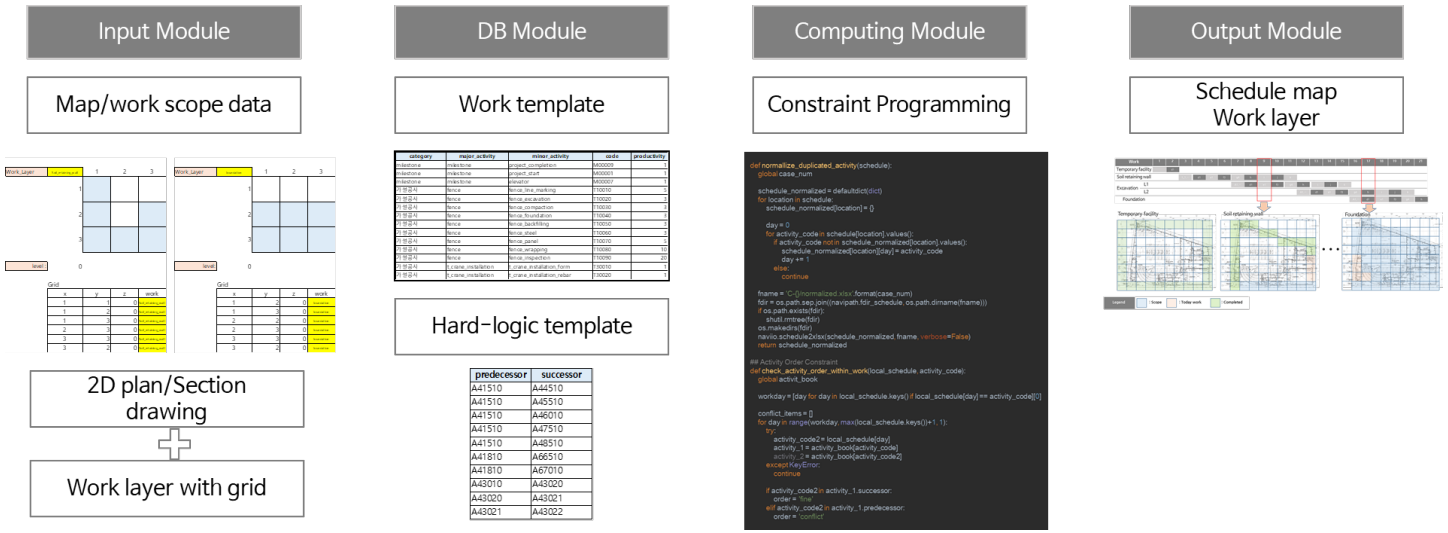

그림 24: 건설공정 내비게이션 구성요소

4.4 내비게이션 작동방법

건설공정 내비게이션은 [그림25]와 같이 계획단계와 수행단계로 구분하여 작동방법을 설명할 수 있다. 계획단계에서 관리자는 현장의 도면을 검토하면서 업무레이어를 이용해 세부공종별로 평면에 포함된 범위를 설정하고, 전체일정 기간을 설정한다. 관리자는 작업별 생산성(1일 작업 그리드수), 작업별 투입 팀수(그리드 수 증가), 선행작업완료 거리를 조정할 수 있다. 또한, 업무템플릿에 정의된 세부작업(자재이동, 검측, 공법추가, 작업순서 변경)을 수정하여 공법의 특성에 맞게 작업내용를 정의할 수 있다. 그리고 선후관계를 수정하거나 추가, 삭제하여 작업의 순서를 프로젝트 전략에 맞게 조정할 수 있다.

각 공종별 관리자에 의해 입력된 그리드별로 배정된 업무는 업무템플릿과 선후관계템플릿으로 작업리스트를 생성하고 작업순서를 배열한다. 그리고 소프트로직 알고리즘의 제약프로그램으로 그리드간 작업 순서를 배열하고 일정을 조정한다.

연산된 작업일정은 일일단위로 상세하게 수립하고 업무레이어에 작업별 작업 위치를 시각화한다. 수립된 결과물을 바차트와 업무레이어 표시된 작업 영역을 일일 단위로 검토하여, 초기관리 기준 공정계획을 확정한다.

수행단계에는 계획된 일정에 따라 작업을 진행하고, 완료된 작업을 업무레이어를 활용하여 입력한다. 이때 각 작업의 담당 협력사별로 업무레이어를 제공하여 작업계획을 소통할 수 있고, 완료된 결과를 작업자가 표시하여 현황을 업데이트 할 수 있다.

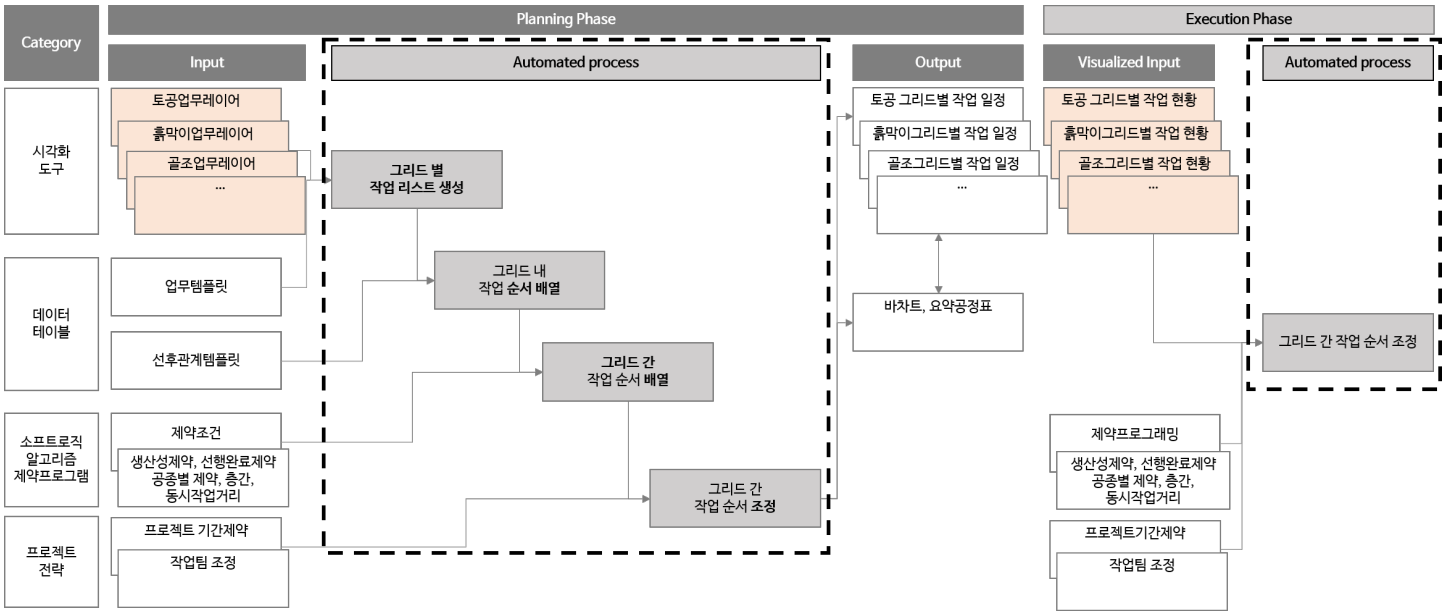

그림 25: 건설공정 내비게이션의 자동화된 프로세스

예를 들어, [그림26], [그림27]과 같이 흙막이공사와 토공사, 기초공사의 업무범위를 설정하는 것으로 각 그리드에 세부작업이 선후관계 템플릿에 정의된 순서로 배정된다.

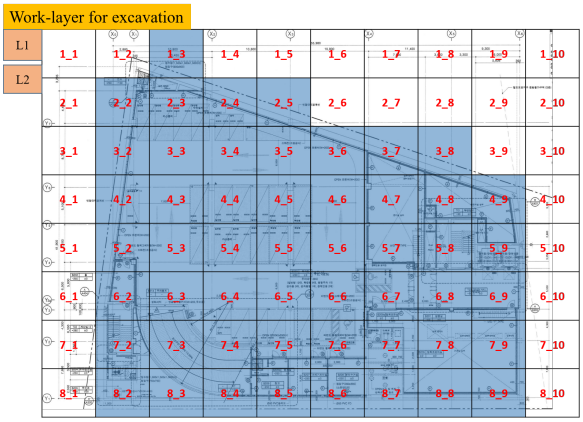

그림 26: 토공사, 기초공사 범위

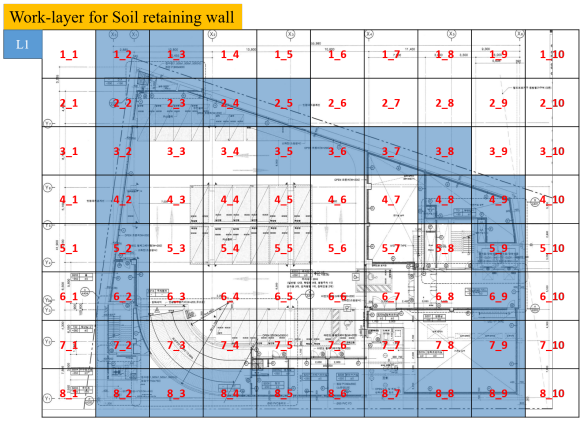

그림 27: 흙막이 업무 범위

각 작업은 제약프로그램에 의해 작업순서가 조정되어 [그림28] 과 같이 바차트 공정표가 자동으로 작성되고 각 일자에 해당하는 업무레이어별 세부작업과 작업범위가 그리드의 색상으로 표시된다.

정리하면, 관리자는 평면에 업무레이어별로 범위설정을 하는 것과 일부 제약기준을 정의하는 것만으로 의사소통이 용이하게 시각화된 공정계획 수립할 수 있다. 그리고 현황을 반영하여 잔여 작업 상황에 맞는 수정된 공정계획을 얻을 수 있다.

그림 28: 바차트와 일일작업 시각화

제 5 장

프로토타입 검증과 투자가치 검토

프로토타입 검증은 데이터의 생성과 연산과정에 필요한 내비게이션의 입력모듈과 DB모듈를 활용한 연산모듈을 엑셀 매크로, VBA(Visual Basic Applications)와 파이썬 프로그래밍을 이용하여 프로토타입 프로그램으로 개발하였다. 그리고 프로젝트 초기의 작업 일부를 가지고 구현한 프로그램의 결과를 검증하였다.

투자가치 검토는 건설공정 내비게이션을 상용화 가능한 프로그램으로 개발하는데 필요한 비용을 개략적으로 산출하고, 중급기술자 이상의 공종별 현장 관리자가 공정계획 업무를 기존의 방식으로 수행하는데 투입되는 시간과 인건비용을 산정하여, 투자비용 회수 기간을 검토하였다.

5.1 프로토타입 개발과 구현 검증

5.1.1 입력 모듈

프로토타입의 입력 모듈은 업무별로 평면에 업무범위를 표시하여 작업데이터를 얻는 과정을 구현한다. 업무레이어에 배정된 업무를 그리드 위치와 업무배정, 배정된 업무를 작업으로 분개하는 과정은 엑셀의 매크로를 활용한다. 업무레이어에는 그리드로 분할된 영역 표시도구와 업무정보를 가진다. 업무를 포함하는 그리드에 범위를 설정한다.

1) [그림29]과 같이 업무별로 그리드를 가지는 레이어로 업무범위를 선택하면 선택된 그리드의 x,y,z 값을 가져오고, 각 정수값을 병합하여 그리드 값을 생성한다.

Work_Layer	T100	fence		
	Grid	1	2	3
	1			
	2			
	3			

level :	0

Grid

x	y	z	work code	work
1	1	0	T100	fence
1	2	0	T100	fence
1	3	0	T100	fence
2	1	0	T100	fence
2	3	0	T100	fence
3	1	0	T100	fence
3	2	0	T100	fence
3	3	0	T100	fence

그림 29: 입력 모듈

2) [그림30]그리드별 업무는 엑셀의 매크로를 이용하여, 업무템플릿에 정의된 세부 작업으로 분개된다. 각 작업은 하루에 수행할 수 있는 생산성 값과 선행완료거리 제약값을 가진다.

x	y	z	major_activity	wp_name_en
1	1	0	T100	fence
1	2	0	T100	fence
1	3	0	T100	fence
2	1	0	T100	fence
2	3	0	T100	fence
3	1	0	T100	fence
3	3	0	T100	fence
3	2	0	T100	fence
4	4	0	M000	Milestone

X

wp_code	wp_name_en	act_id	act_name_en	productivity	predecessor_dist
M000	Milestone	M00001	Milestone-project_Start	1	0
M000	Milestone	M00009	Milestone-project_completion	1	5
T100	fence	T10010	fence-line_marking	10	5
T100	fence	T10020	fence-excavation	3	1
T100	fence	T10030	fence-compaction	3	1
T100	fence	T10040	fence-foundation	3	0
T100	fence	T10050	fence-backfilling	3	0
T100	fence	T10060	fence-steel	3	0
T100	fence	T10070	fence-panel	5	2
T100	fence	T10080	fence-inspection	6	3

x	y	z	major_act	wp_name_en	act_id	act_name_en	productivity	predecessor_dist
1	1	0	T100	fence	T10010	fence-line_marking	10	5
1	1	0	T100	fence	T10020	fence-excavation	3	1
1	1	0	T100	fence	T10030	fence-compaction	3	1
1	1	0	T100	fence	T10040	fence-foundation	3	0
1	1	0	T100	fence	T10050	fence-backfilling	3	0
1	1	0	T100	fence	T10060	fence-steel	3	0
1	1	0	T100	fence	T10070	fence-panel	5	2
1	1	0	T100	fence	T10080	fence-inspection	6	3
1	2	0	T100	fence	T10010	fence-line_marking	10	5
1	2	0	T100	fence	T10020	fence-excavation	3	1
1	2	0	T100	fence	T10030	fence-compaction	3	1
1	2	0	T100	fence	T10040	fence-foundation	3	0
1	2	0	T100	fence	T10050	fence-backfilling	3	0
1	2	0	T100	fence	T10060	fence-steel	3	0
1	2	0	T100	fence	T10070	fence-panel	5	2
1	2	0	T100	fence	T10080	fence-inspection	6	3
1	3	0	T100	fence	T10010	fence-line_marking	10	5
1	3	0	T100	fence	T10020	fence-excavation	3	1
1	3	0	T100	fence	T10030	fence-compaction	3	1
1	3	0	T100	fence	T10040	fence-foundation	3	0
1	3	0	T100	fence	T10050	fence-backfilling	3	0
1	3	0	T100	fence	T10060	fence-steel	3	0
1	3	0	T100	fence	T10070	fence-panel	5	2
1	3	0	T100	fence	T10080	fence-inspection	6	3
2	1	0	T100	fence	T10010	fence-line_marking	10	5
2	1	0	T100	fence	T10020	fence-excavation	3	1
2	1	0	T100	fence	T10030	fence-compaction	3	1
2	1	0	T100	fence	T10040	fence-foundation	3	0
2	1	0	T100	fence	T10050	fence-backfilling	3	0
2	1	0	T100	fence	T10060	fence-steel	3	0
2	1	0	T100	fence	T10070	fence-panel	5	2
2	1	0	T100	fence	T10080	fence-inspection	6	3
2	3	0	T100	fence	T10010	fence-line_marking	10	5
2	3	0	T100	fence	T10020	fence-excavation	3	1
2	3	0	T100	fence	T10030	fence-compaction	3	1
2	3	0	T100	fence	T10040	fence-foundation	3	0
2	3	0	T100	fence	T10050	fence-backfilling	3	0
2	3	0	T100	fence	T10060	fence-steel	3	0
2	3	0	T100	fence	T10070	fence-panel	5	2
2	3	0	T100	fence	T10080	fence-inspection	6	3
3	1	0	T100	fence	T10010	fence-line_marking	10	5
3	1	0	T100	fence	T10020	fence-excavation	3	1
3	1	0	T100	fence	T10030	fence-compaction	3	1
3	1	0	T100	fence	T10040	fence-foundation	3	0
3	1	0	T100	fence	T10050	fence-backfilling	3	0
3	1	0	T100	fence	T10060	fence-steel	3	0
3	1	0	T100	fence	T10070	fence-panel	5	2
3	1	0	T100	fence	T10080	fence-inspection	6	3
3	3	0	T100	fence	T10010	fence-line_marking	10	5
3	3	0	T100	fence	T10020	fence-excavation	3	1
3	3	0	T100	fence	T10030	fence-compaction	3	1
3	3	0	T100	fence	T10040	fence-foundation	3	0
3	3	0	T100	fence	T10050	fence-backfilling	3	0
3	3	0	T100	fence	T10060	fence-steel	3	0
3	3	0	T100	fence	T10070	fence-panel	5	2
3	3	0	T100	fence	T10080	fence-inspection	6	3
3	2	0	T100	fence	T10010	fence-line_marking	10	5
3	2	0	T100	fence	T10020	fence-excavation	3	1
3	2	0	T100	fence	T10030	fence-compaction	3	1
3	2	0	T100	fence	T10040	fence-foundation	3	0
3	2	0	T100	fence	T10050	fence-backfilling	3	0
3	2	0	T100	fence	T10060	fence-steel	3	0
3	2	0	T100	fence	T10070	fence-panel	5	2
3	2	0	T100	fence	T10080	fence-inspection	6	3
4	4	0	M000	Milestone	M00001	Milestone-project_Start	1	0
4	4	0	M000	Milestone	M00009	Milestone-project_completion	1	5

그림 30: 그리드별 업무의 상세 작업 분개

5.1.2 연산 모듈

연산모듈은 파이썬을 이용하여 그리드별 작업을 제약조건을 가지고 일정을 배정한다. 파이선 프로그램에 주요 작동 모듈은 [표11]와 같이 init과 [표12]와 같은 update로 구성된다. 업무템플릿과 선후관계템플릿의 데이타를 변환하는 기능과 같은 보완 기능은 별도의 모듈로 프로그램하였다. init모듈은 엑셀파일을 읽어서 선후관계템플릿에 정의된 작업 간 순서를 고려하여 제약조건 반영 이전의 데이터파일을 생성한다. update모듈은 init의 데이터를 생산성 제약과 선행완료거리 제약을 작동하여 날짜별, 그리드별로 작업이 이동할 때마다 새로 제약조건을 반복하여 작업 일정을 배정한다.

표 11: 연산 모듈 - init

함수	함수 내용
init_activity_book	· 엑셀의 업무템플릿을 딕셔너리로 변환하여 activity book.pk로 저장
set_orders_in_activity_book	· activitybook.pk의 값을선후관계템플릿에 순서대로 정렬하고 순환오류 검토
define_works	· 그리드와 작업이 배정된 엑셀파일을 딕셔너리로 저장
initiate_project	· defineworks에서 만든 work를 project.pk파일로 저장
initial_schedule	· 위 function을작동하는함수

표 12: 연산 모듈 update

함수	함수 내용
import_schedule	· init모듈의 생성파일을 불러옴
normallize_duplicated _activity	· 그리드별 작업의 제약조건 반영 전 일정생성
check_activity_order _within_work	· 그리드별 작업순서 점검
reorder_activity	· 작업 순서 보정
update_order_in _local_schedule	· 그리드별 작업순서 오류 검출
activity_order_constraint	· 그리드별 작업순서 보정
find_influence_locations	· 작업의 선행완료거리 영향 그리드 리스트 생성
check_pre_dist	· 작업의 선행완료 거리 값 추출
activity_predecessor_ completion_constraint	· 선행완료 거리 영향 내 작업과 선후관계 검토 후 작업 일정 밀기
compress_schedule	· 불필요하게 밀린 작업당기기
update	· update모듈 내 함수 반복 수행

5.1.3 프로토타입 프로그램 검증과 시각화

프로토타입 프로그램은 CSM기법과 이를 기반으로 하는 소프트로직 알고리즘으로 작업 가능한 일정을 자동으로 산출하고 산출된 데이터를 가지고 시각화하였다.

1) [그림31]과 같이 상세 작업은 작업간 선후관계를 가지고 각 작업은 하루에 수행할 수 있는 생산성 값과 선행완료거리 제약값을 가진다. 프로젝트 시작과 종료 마일스톤으로 모든 작업에 선후연결 작업을 가지도록 하였다. 작업의 생산성은 1 - 10까지의 정수를 배정하였다. 작업의 특성상 동시에 넓은 작업이 가능한 작업에 높은 수를, 반대의 경우 낮은 수를 배정하였다. 선행완료거리 값은 0 - 5까지의 정수를 배정하였다. 작업이 시작하기 위해 선행작업의 완료 요구 면적이 큰 작업에 높은 수를, 반대의 경우 낮은 수를 배정하였다.

작업선후관계

predecessor	successor	predecessor	successor
M00001	T10010	Milestone-project_Start	fence-line_marking
T10010	T10020	fence-line_marking	fence-excavation
T10020	T10030	fence-excavation	fence-compaction
T10030	T10040	fence-compaction	fence-foundation
T10040	T10050	fence-foundation	fence-backfilling
T10050	T10060	fence-backfilling	fence-steel
T10060	T10070	fence-steel	fence-panel
T10070	T10080	fence-panel	fence-inspection
T10080	M00009	fence-inspection	Milestone-project_completion

작업생산성 & 선행완료거리

act_id	act_name_en	productivity	predecessor_dist
M00001	Milestone-project_Start	1	0
M00009	Milestone-project_completion	1	5
T10010	fence-line_marking	10	5
T10020	fence-excavation	3	1
T10030	fence-compaction	3	1
T10040	fence-foundation	3	0
T10050	fence-backfilling	3	0
T10060	fence-steel	3	0
T10070	fence-panel	5	2
T10080	fence-inspection	6	3

그림 31: 선후관계템플릿과 업무템플릿 적용

2) [그림30]에서 분개된 작업을 가지고 init 모듈을 거친 각 그리드에 배정된 세부작업은 [그림32]과 같이 그리드 별 작업이 순서대로 배정되었다.

		0	1	2	3	4	5	6	7
그리드별 작업정렬 & 작업선후제약	1_1_0	T10010	T10020	T10030	T10040	T10050	T10060	T10070	T10080
	1_2_0	T10010	T10020	T10030	T10040	T10050	T10060	T10070	T10080
	1_3_0	T10010	T10020	T10030	T10040	T10050	T10060	T10070	T10080
	2_1_0	T10010	T10020	T10030	T10040	T10050	T10060	T10070	T10080
	2_3_0	T10010	T10020	T10030	T10040	T10050	T10060	T10070	T10080
	3_1_0	T10010	T10020	T10030	T10040	T10050	T10060	T10070	T10080
	3_2_0	T10010	T10020	T10030	T10040	T10050	T10060	T10070	T10080
	3_3_0	T10010	T10020	T10030	T10040	T10050	T10060	T10070	T10080
	4_4_0	M00001	M00009						

그림 32: 그리드별 작업과 순서 배정된 초기 공정표

3) [그림33]과 Update 모듈 내 order함수와 선행완료거리제약과 관련한 함수를 거친 공정표가 제약조건에 맞게 수정되는 것을 확인하였다.

		0	1	2	3	4	5	6	7	8	9
작업선후제약 + 선행완료거리 제약	1_1_0	------	T10010	T10020	T10030	T10040	T10050	T10060	T10070	T10080	
	1_2_0	------	T10010	T10020	T10030	T10040	T10050	T10060	T10070	T10080	
	1_3_0	------	T10010	T10020	T10030	T10040	T10050	T10060	T10070	T10080	
	2_1_0	------	T10010	T10020	T10030	T10040	T10050	T10060	T10070	T10080	
	2_3_0	------	T10010	T10020	T10030	T10040	T10050	T10060	T10070	T10080	
	3_1_0	------	T10010	T10020	T10030	T10040	T10050	T10060	T10070	T10080	
	3_2_0	------	T10010	T10020	T10030	T10040	T10050	T10060	T10070	T10080	
	3_3_0	------	T10010	T10020	T10030	T10040	T10050	T10060	T10070	T10080	
	4_4_0	M00001	------	------	------	------	------	------	------	------	M00009

그림 33: 작업순서와 선행완료거리 제약 적용 공정표

4) [그림34]과 Update 모듈 내 order함수와 생산성제약과 관련한 함수를 거친 공정표가 제약조건에 맞게 수정되는 것을 확인하였다. 이때 선행완료거리제약 관련한 함수는 적용하지 않도록 하여 생산성제약 함수의 작동을 확인하였다.

		0	1	2	3	4	5	6	7	8	9
작업선후제약 + 생산성제약	1_1_0	T10010	------	------	T10020	T10030	T10040	T10050	T10060	T10070	T10080
	1_2_0	T10010	------	------	T10020	T10030	T10040	T10050	T10060	T10070	T10080
	1_3_0	T10010	------	T10020	T10030	T10040	T10050	T10060	T10070	T10080	
	2_1_0	T10010	------	T10020	T10030	T10040	T10050	T10060	T10070	T10080	
	2_3_0	T10010	------	T10020	T10030	T10040	T10050	T10060	T10070	T10080	
	3_1_0	T10010	T10020	T10030	T10040	T10050	T10060	T10070	T10080		
	3_2_0	T10010	T10020	T10030	T10040	T10050	T10060	T10070	T10080		
	3_3_0	T10010	T10020	T10030	T10040	T10050	T10060	T10070	T10080		
	4_4_0	M00001	M00009								

그림 34: 작업순서와 생산성 제약 적용 공정표

5) [그림35]과 Update 모듈 내 프로그래밍한 모든 함수를 거친 공정표가 제약조건에 맞게 수정되었다.

		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
작업선후제약 + 생산성제약 + 선행완료거리 제약	1_1_0	------	T10010	------	------	T10020	------	T10030	T10040	T10050	T10060	------	T10070	T10080		
	1_2_0	------	T10010	------	------	T10020	------	T10030	T10040	T10050	T10060	------	T10070	------	T10080	
	1_3_0	------	T10010	------	T10020	------	T10030	T10040	T10050	T10060	------	------	T10070	------	T10080	
	2_1_0	------	T10010	------	T10020	------	T10030	T10040	T10050	T10060	------	T10070	------	T10080		
	2_3_0	------	T10010	------	T10020	------	T10030	T10040	T10050	T10060	------	T10070	------	T10080		
	3_1_0	------	T10010	T10020	------	T10030	T10040	T10050	T10060	------	------	T10070	------	T10080		
	3_2_0	------	T10010	T10020	------	T10030	T10040	T10050	T10060	------	------	T10070	------	T10080		
	3_3_0	------	T10010	T10020	------	T10030	T10040	T10050	T10060	------	------	T10070	------	T10080		
	4_4_0	M00001	------	------	------	------	------	------	------	------	------	------	------	------	------	M00009

그림 35: 작업순서, 생산성 제약, 선행완료거리 제약 적용 공정표

6) [그림36]과 같이 연산모듈에서 나온 결과를 가지고 입력모듈에서 활용할 엑셀을 활용하여 각 날짜와 작업별로 그리드에 표시하였다.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1_1_0	------	T10010	------	------	T10020	------	T10030	T10040	T10050	T10060	------	T10070	T10080		
1_2_0	------	T10010	------	------	T10020	------	T10030	T10040	T10050	T10060	------	T10070	------	T10080	
1_3_0	------	T10010	------	T10020	------	T10030	T10040	T10050	T10060	------	------	T10070	------	T10080	
2_1_0	------	T10010	------	T10020	------	T10030	T10040	T10050	T10060	------	T10070	------	T10080		
2_3_0	------	T10010	------	T10020	------	T10030	T10040	T10050	T10060	------	T10070	------	T10080		
3_1_0	------	T10010	T10020	------	T10030	T10040	T10050	T10060	------	------	T10070	------	T10080		
3_2_0	------	T10010	T10020	------	T10030	T10040	T10050	T10060	------	------	T10070	------	T10080		
3_3_0	------	T10010	T10020	------	T10030	T10040	T10050	T10060	------	------	T10070	------	T10080		
4_4_0	M00001	------	------	------	------	------	------	------	------	------	------	------	------	------	M00009

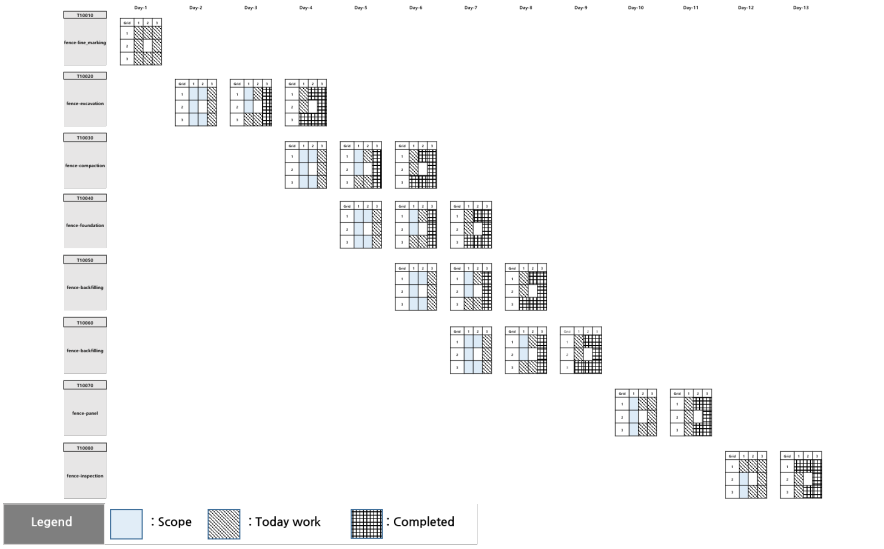

그림 36: 연산모듈 결과를 활용한 시각화 적용 공정표

7) [그림37]은 초기 5일까지의 시각화된 공정계획이다. [그림38]는 실제 수행한 4일까지의 작업을 임의로 계획과 상이하게 진행하여 표시하고 5일 이후 작업을 연산모듈을 거쳐 수정된 계획을 얻은 결과이다. 진행 작업과 잔여 작업의 현황이 달라짐에 따라 잔여계획이 변경된 것을 확인할 수 있다.

	0	1	2	3	4	5	6
1_1_0	------	T10010	------	------	T10020	------	T10030
1_2_0	------	T10010	------	------	T10020	------	T10030
1_3_0	------	T10010	------	T10020	------	T10030	T10040
2_1_0	------	T10010	------	T10020	------	T10030	T10040
2_3_0	------	T10010	------	T10020	------	T10030	T10040
3_1_0	------	T10010	T10020	------	T10030	T10040	T10050
3_2_0	------	T10010	T10020	------	T10030	T10040	T10050
3_3_0	------	T10010	T10020	------	T10030	T10040	T10050
4_4_0	M00001	------	------	------	------	------	------

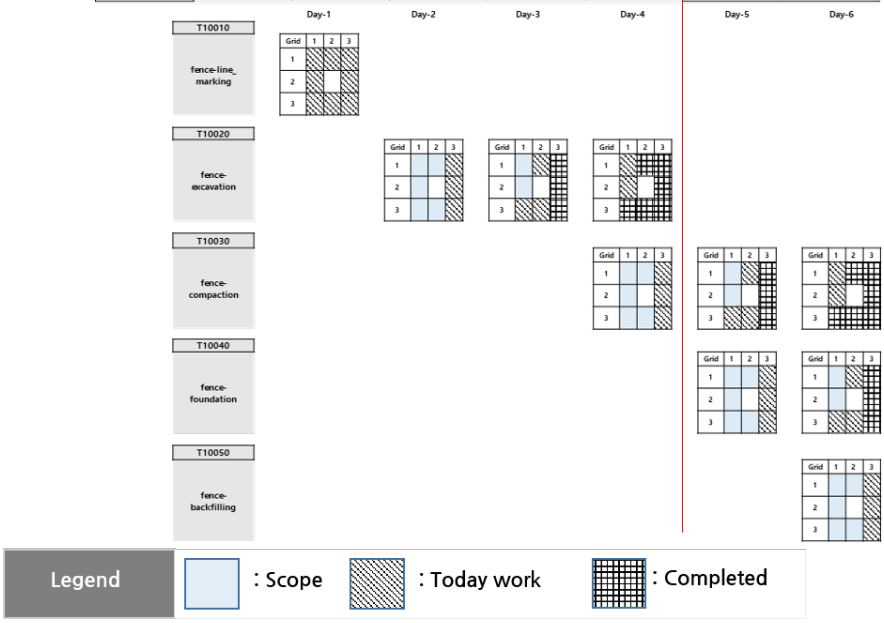

그림 37: 초기 계획

	0	1	2	3	4	5	6
1_1_0	------	T10010	------	T10020	T10030	------	------
1_2_0	------	T10010	------	T10020	T10030	T10040	T10050
1_3_0	------	T10010	------	T10020	T10030	T10040	T10050
2_1_0	------	T10010	------	T10020	T10030	T10040	T10050
2_3_0	------	T10010	T10020	------	------	------	T10030
3_1_0	------	T10010	T10020	------	------	T10030	T10040
3_2_0	------	T10010	T10020	------	------	T10030	T10040
3_3_0	------	T10010	T10020	------	------	------	T10030
4_4_0	M00001	------	------	------	------	------	------

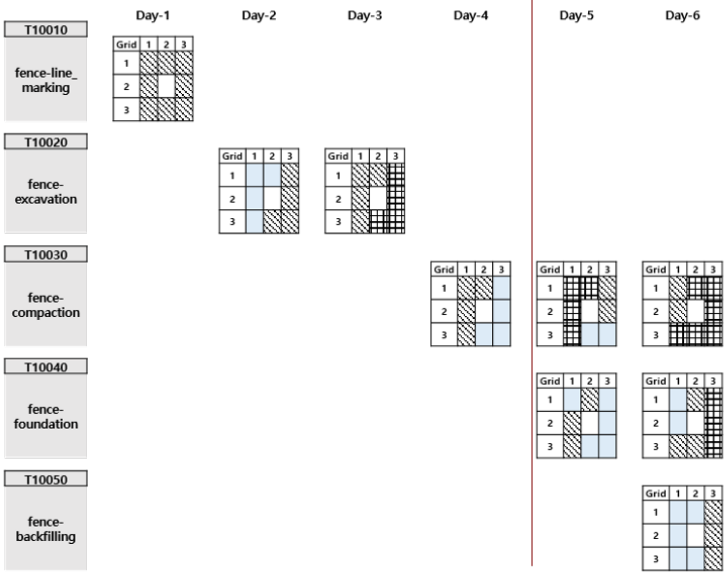

그림 38: 업데이트 현황과 수정된 공정계획

건설공정 내비게이션의 입력모듈과 DB모듈를 활용한 연산모듈을 구현하는 프로토타입 프로그램을 가지고 업무범위설정과 작업분개, 작업의 일정배정을 자동으로 수행하는 것을 확인하였다. 작업순서 제약을 준수하고, 작업의 생산성과 작업이 시작하기 위한 선행완료거리 제약에 따라 작업 가능한 일정이 배정되었다. 그리고 작업현황이 계획과 달리 수행되어 잔여 작업 현황이 달라짐에 따라 잔여계획이 기존과 다르게 자동으로 수정되었다.

이와 같은 실험으로 본 연구에서 제안한 CSM과 이를 기반한 소프트로직 알고리즘이 공정계획을 자동화하고 시각화할 수 있는 데이터를 생성할 수 있음을 증명하였다.

5.2 프로그램 개발 투자가치 검토

프로그램 개발 조건을 갖춘 것을 전제로 프로그램의 개발에 필요한 비용을 산출하기 위해 공정관리 컨설팅 업체, 프로그램 개발 업체와 미팅하였다. 개략적인 견적으로 고급 프로그래머 3명이 최소 3개월 투입이 요구되는 것으로 판단하였다. 인건비를 기준으로 산출하여 인당 월 1,000만원으로 계산 시 9,000만원이 예상되며, 연관된 라이센스 구입과 프로그램 구현 범위의 변경, 각종 리스크에 대비하는 예비비용으로 9,000만원을 확보하면, 1.8억의 예산이 필요한 것으로 산출하였다.

이에 편익의 계산으로, 기존 방식으로 공정관리에 투입되는 인건비를 기준으로 산정하였다. 건설공정 내비게이션으로 얻을 수 있는 초기단계 공정계획의 간소화 시간과 운영 단계에 주간, 월간 계획의 수립의 자동화가 가능한 것으로 산정 기준을 정립하였다.

기존 공정관리 방식으로 CPM을 활용하여 업무 범위의 누락없이 논리적으로 공정계획을 수립하는 데에는 일반적인 시공 프로젝트는 1개월의 기간이 소요된다. 이를 수행하는데 공정관리 전문인력의 투입된 시간은 최소 30일을 기준으로 산정하였다. 그리고 각 공종별 수행계획을 수립하는데 각 공종의 담당자인 건축, 설비, 전기 공사 관리자 3명이 투입하는 시간은 초기 1개월 중 계획에 투입하는 시간을 업무 비중을 20%로 가정하여 총 6일로 산정하였다.

운영단계에는 주간단위와 월간단위로 상세 공정계획을 주기적으로 수립한다. 주간단위 공정계획에 투입되는 시간과 월간단위 공정계획에 투입되는 시간은 15명의 현장의 공사팀장을 대상으로 인터뷰하여 평균 주간계획은 2시간/주, 5시간/월로 산정하였다.

주간단위 계획은 프로젝트 기간을 1년으로 고려할 때 52개주, 2시간,

3명의 담당자가 총 연간 312시간이 소요된다. 1일 근무시간을 8시간으로 환산하면 39일이다. 월간단위 계획은 프로젝트 기간을 1년으로 고려할 때 12개월, 5시간, 3명의 담당자가 총 연간 180시간이 소요된다. 1일 근무시간을 8시간으로 환산하면 23일이다. 즉, 주간단위 계획과 월간단위 계획에 투입되는 품은 총 62인일의 투입이 요구된다.

이를 비용으로 환산하기 위한 일당인건비를 계산 편의를 위해 30만원(월급과 각종수당, 복지비 등 포함)으로 일괄 산정한다. 초기 계획에 투입된 공정관리자 30일과 공종별 관리자 3명이 6일 투입으로, 총 900만원과 540만원으로 1,440만원으로 산정할 수 있다. 주간과 월간 계획에는 연단위로 62인일에 일단가 30만원의 곱으로 1,860만원을 산정할 수 있다. 매년 1개 현장에서 한번의 초기 계획과 1년간의 운영을 진행하는 것으로 간주할 때, 3,300만원의 인건비가 투입되고 있는 것으로 볼 수 있다.

건설공정 내비게이션 사용 시 공정계획에 투입되는 비용을 산정하면, 초기 계획에 공정관리자가 30일간 투입하는 인건비용과 그리고 주간, 월간 단위의 계획은 자동화로 전액 제외가 가능하다. 그러나 각 공종의 관리자가 수행계획에 투입하는 시간은 필수적이다. 따라서, 초기 계획시 투입되는 공종별 관리자 3명이 6일간 투입되는 비용만을 감안하여 총 540만원으로 산정할 수 있다. 결국, 건설공정 내비게이션 사용 시 연간 1개 현장별로 2,760만원의 인건비 절감이 가능하다.

중견급 건설사인 C사의 경우 연간 평균 30개 현장을 운영을 기준으로 산정할 때, 연간 8억2천800만원의 절감 효과를 기대할 수 있다. 앞서 개략으로 산출한 프로그램 구현에 필요할 것으로 예상되는 비용 약 1.8억원을 투자한다면, 프로그램 정착 이후 1분기 내에 원금회수가 가능한 것으로 판단할 수 있다[표13].

이는 단순 인건비로 산출한 금액이며, 해당 인력이 공정표를 작성하

고 수정하는 시간에 품질과 공법개선과 같이 부가가치를 높이는 업무를 수행하여, 얻을 수 있는 추가 이익은 제외하고 판단한 수치이다.

국내의 중소 건설사가 해당 프로그램을 구매하거나 무상으로 보급받는다면, 과학적인 공정관리로 생산성 향상과 자원절약으로 건설분야에서 국가 경쟁력을 높이는 데 기여할 수 있을 것이다.

표 13: 절감 비용 및 투자 비용

단계	구분	산출근거	예상 비용(만원)
초기 계획 단계	관리 기준 공정계획 투입인건비	공정관리자 1명 30일 100% 투입 - 1명 * 30일 * 30만원/일	900
		중급이상 관리자 3명(건축, 설비, 전기) 1개월 20% 투입 - 3명 * 6일 * 30만원/일	540
	소계(1)		1440
운영 단계	주간 공정계획 투입인건비	중급이상 관리자 3명 1개월 20% 투입 52주 * 2시간 = 39일 - 39일 * 30만원/일	1170
	월간 공정계획 투입인건비	중급이상 관리자 3명(건축, 설비, 전기) 12개월 * 5시간 = 23일 - 23일 * 30만원/일	690
	소계(2)		1860
연간		(1) + (2) = 총계(3)	3300

공정 내비게이션 초기 계획 투입 인건비(4)	중급이상 관리자 3명(건축, 설비, 전기) 1개월 20% 투입 3명 * 6일 * 30만원/일	540
1개 현장 연간 절감액(5)	(3) - (4)	2760
30개 현장 절감액(6)	(5) * 30개 현장	82,800

개발투자비	개발비(7)	고급 프로그램 엔지니어 3명 3개월 투입 월1천만원 * 3명 * 3개월	9,000
	라이선스 비용 및 예비비(8)	(7) * 100%	9,000
	(7) + (8) = 소계(9)		18,000

제 6 장

결론

6.1 연구 요약

본 연구는 기존 공정관리 방법이 가지고 있는 과학적 공정관리 전문성의 요구로 인한 중소건설사의 적용 한계, 공정계획 · 수정 및 시각화에 수작업의 높은 의존도와 같은 문제점을 선행 연구와 실무 현황을 바탕으로 제시하였다.

이를 해결하기 위해 차량용 내비게이션과 같이 사용이 쉽고 시각화하는 건설공정 내비게이션 개발방법을 고안하였다. 건설공정 내비게이션은 그리드로 공간을 세세하게 분할한 업무레이어를 활용하여, 업무별 범위를 설정하는 것 만으로 공정계획을 자동으로 수립하고 계획된 작업 영역을 시각화하는 도구이다. 이를 구현하기 위해 기존과 차별화된 공정관리 기법인 CSM을 제안하고, 그리드간 작업의 순서를 자동으로 배정하는 소프트로직 알고리즘을 개발하였다. 이렇게 개발된 요소를 가지고 건설공정 내비게이션의 구성요소와 작동방법을 설명하였다.

그리고 업무의 범위설정으로 부터 상세작업 분개과정은 엑셀의 매크로, VBA를 활용하고, 소프트로직 알고리즘을 구현에 파이썬을 이용하여 프로토타입 프로그램을 개발하였다. 이를 가지고, 그리드별로 포함된 작업을 작업순서와 생산성, 선행작업의 거리 제약 조건에 맞게 일정을 배정하는 것을 확인하였다. 마지막으로 프로그램 개발 투자가치를 검토하여 개발 타당성을 검증하였다.

위와 같은 연구로 업무별로 범위를 설정하는 것만으로 공정계획을 자동으로 수립하고 작업 진행현황에 따라 공정계획을 자동으로 수정할 수 있는 건설공정 내비게이션 구현 가능성과 개발 타당성을 확인하였다.

6.2 학술적 기여

이번 연구는 차별화된 공정관리 기법인 CSM을 제안하고 이를 기반으로하는 프로토타입 프로그램의 구현으로서 의의를 가진다. 이는 공정관리에 전문성 없이도 공정계획을 상세하게 수립하고 관리하게 지원함으로써 대형건설사는 물론, 중소건설사도 조직구축과 투자의 부담없이 공정관리 수준을 높일 수 있는 가능성을 기대할 수 있다. 또한 주기적으로 수행하는 상세공정계획 수립 업무를 자동으로 하여, 관리자가 프로젝트 가치를 높일 수 있는 업무에 시간을 투자할 수 있을 것이다. 4차 산업혁명과 코로나19로 인해 빠르게 변화하는 건설환경 속에서 건설관리 패러다임을 변화하는 도구가 될 수 있을 것이다.

6.3 향후 연구

본 연구는 그리드의 규격, 생산성제약과 선행완료거리 제약에 사용된 값은 임의 값으로 설정하였다. 그리드의 크기 변화에 따라 생산성과 선행완료거리 값이 변동되어야 하고, 컴퓨팅 환경에 따라 연산속도를 고려한 적정 그리드 규격에 대한 연구와 실험이 필요하다.

그리고 공정 내비게이션의 계획과 운영과정을 실제 프로젝트에 적용하여 작업의 생산성과 선행완료거리 제약의 실제적인 데이터를 축적하는 연구가 요구된다. 그리고 운영과정에 생성성 제약 값과 선행완료제약 값을 업데이트하여 잔여 계획에 반영하는 연구가 필요하다. 그리고 층간 연결 조건과 자원의 연속성을 고려하여 공정 내비게이션이 제공하는 공정 계획이 비용이나 자원 측면에서 최적화 알고리즘을 추가적으로 보완하는 연구가 필요하다.

참고 문헌

[1] S. Chi, "4 차 산업혁명 시대-미래 도시건설 전망," *Construction Engineering and Management*, vol. 18, no. 3, pp. 10–13, 2017.

[2] S. Eom, "4 차산업혁명과 건설산업, 그리고 CM," *Construction Engineering and Management*, vol. 18, no. 4, pp. 3–7, 2017.

[3] Y. Jung and H. Kim, "Patterns and trends in schedule control practice of korean general contractors," *Korean Journal of Construction Engineering and Management*, vol. 12, no. 4, pp. 88–96, 2011.

[4] W. Kim, "A knowledge-based approacch to plant construction process planning," *Journal of Intelligence and Information Systems*, vol. 7, no. 1, pp. 81–95, 2001.

[5] M. Jang and Y. Lee, "Time management status of small contractors and suggestions on education for time management in colleges," *Asia-pacific Journal of Multimedia services convergent with Art, Humanities and Sociology*, vol. 6, pp. 413–422, 2016.

[6] H. Lee, J. Kim, M. Park, and H. Ryu, "Aggressive buffer control (ABC) for reliable construction schedule management," *Journal of the architectural institute of Korea : Structure & Construction*, vol. 22, no. 10, pp. 137–146, 2006.

[7] Y. Huh, H. Seo, and E. Kim, "Analysis of scheduling technique applied by medium-sized construction company in Busan," *Journal of the Regional Association of Architectural Institute of Korea*, vol. 17, no. 3, pp. 137–144, 2015.

[8] K. Kim, S. Kim, and J. Kim, "CPM bar chart technique for construction scheduling," *Korean Institute Of Construction Engineering and Management*, vol. 5, no. 5, pp. 135–142, 2004.

[9] R. M. Reda, "RPM: Repetitive project modeling," *Journal of Construction Engineering and Management*, vol. 116, no. 2, pp. 316–330, 1990.

[10] L. J. B. Ragel *et al.*, "Limitations of PERT/CPM in construction management planning: Inputs to mathematics in architecture education," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 10, pp. 5218–5223, 2021.

[11] S. Jang, M. Park, and H. Lee, "A study on schedule management in construction project using the dependency structure matrix(DSM)," 2007.

[12] N. K. Rad, *Project Scheduling Rules, 2nd edition*. bookboon.com, 2013.

[13] O. Stradal and J. Cacha, "Time space scheduling method," *Journal of the Construction Division*, vol. 108, no. 3, pp. 445–457, 1982.

[14] K. Kim, J. Kim, and K. Kim, "A space scheduling model reflecting work area," *Journal of the architectural institute of Korea*, vol. 13, no. 1, pp. 235–245, 1997.

[15] H. Lee, G. Park, M. Park, and H. Ryu, "Construction planning model for using the working space of high-rise building construction," *Journal of the architectural institute of Korea : Structure & Construction*, vol. 26, no. 6, pp. 129–138, 2010.

[16] H. Ryu, "A method of applying work relationships for a linear scheduling model," *Journal of the Korea Institute of Building Construction*, vol. 10, no. 4, 2010.

[17] S. Joo, M. Park, H. Lee, and G. Lee, "TACT productivity management for finish works of residential buildings using productivity achievement ratio (PAR)," *Korean Journal of Construction Engineering and Management*, vol. 16, no. 3, pp. 36–48, 2015.

[18] X. Zou and L. Zhang, "A constraint programming approach for scheduling repetitive projects with atypical activities considering soft logic," *Automation in Construction*, vol. 109, p. 102990, 2020.

[19] V. Faghihi, A. Nejat, K. F. Reinschmidt, and J. H. Kang, "Automation in construction scheduling: a review of the literature," *The International Journal of Advanced Manufacturing Technology*, vol. 81, no. 9, pp. 1845–1856, 2015.

[20] B. Choi, H. Lee, M. Park, and H. Kim, "4D BIM based workspace planning process in building construction project," *Journal of Korean Institute Of Construction Engineering and Management*, vol. 14, no. 5, p. 175, 2013.

[21] J. Choi and H. Ryu, "Application and effects analysis of BIM(building information modeling) for construction management of a construction field," *Journal of the Korea Institute of Building Construction*, vol. 15, no. 1, pp. 115–121, 2015.

[22] W. Y. Thabet and Y. J. Beliveau, "Hvls: Horizontal and vertical logic scheduling for multistory projects," *Journal of Construction Engineering and Management*, vol. 120, no. 4, pp. 875–892, 1994.

[23] S. Ahn, H. Lee, M. Park, and W. Kim, "Object-oriented modeling of construction operations for schedule-cost integration, based on BIM," *Journal of the architectural institute of Korea : Structure & Construction*, vol. 25, no. 12, pp. 165–174, 2009.

[24] B. de Vries and J. M. Harink, "Generation of a construction planning from a 3d cad model," *Automation in Construction*, vol. 16, no. 1, pp. 13–18, 2007.

[25] S. Song, J. Yang, and T. Myung, "Automated generation of a construction schedule based on the work method template for 4D simulation," *IE interfaces*, vol. 25, no. 2, pp. 216–228, 2012.

[26] K. Oh, G. Park, and J. Song, "A study on the using of BIM data and template for construction progress management," *Journal of the Korea Academia-Industrial cooperation Society*, vol. 17, no. 8, pp. 157–163, 2016.

[27] S. Park, J. Song, and K. Oh, "Development of construction schedule management system based on BIM," *Journal of the architectural institute of Korea : Structure & Construction*, vol. 34, no. 3, pp. 61–68, 2018.

[28] J. Ryu and G. Kim, "A study of 4D simulation system using automatic scheduling process - the focus on steel structural construction," *Journal of the architectural institute of Korea : Structure & Construction*, vol. 25, no. 9, pp. 173–180, 2009.

[29] J. Lee, H. Lee, M. Park, and M. Jung, "A framework integrating cost and schedule based on BIM using IFC," *Korean journal of construction engineering and management*, vol. 14, no. 3, pp. 53–64, 2013.

Abstract

A Study on Program Development for Automated Construction Scheduling and Visualization

Heung Soon CHOI
Graduate School of Practical Engineering
Seoul National University

The 4th industrial revolution technology and the new normal situation caused by the pandemic increase the uncertainty, volatility and complexity of construction projects. The purpose of this study is to develop a schedule management tool that automatically establishes and visualizes a detailed construction plan without professional knowledge and specialized schedule management tools such as a car navigation. To this end, the problems of existing schedule management and previous studies related to scheduling automation and visualization are considered. Considering the existing research results in a complex way, we study a tool development method that can visually express the location and content of work without BIM modeling. To this end, we propose a countable grid scheduling method (CSM) differentiated from the existing ones, and develop a soft logic algorithm that automatically establishes and changes plans. Based on this, the components

and operation method of the construction schedule navigation are explained, and the utility is verified with the work data of some types of work that can be done in the planning stage by implementing a prototype in Python. It is expected that this study will contribute to the improvement of construction project management, which preemptively responds to variable customer demands according to market changes and quickly judges the impact of the introduction of new construction methods.

Keywords : Construction Schedule Management, Countable grid Scheduling Method, Soft logic, Automation, Visualization, Constraint Programming, Grid, Template, Construction Schedule Navigation

Student Number : 2020-25279