REPUBLIC OF TURKEY

YILDIZ TECHNICAL UNIVERSITY

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

DEVELOPMENT OF A BIM BASED DELAY ANALYSIS METHOD IN CONSTRUCTION INDUSTRY

Murat ÇEVİKBAŞ

DOCTOR OF PHILOSOPHY THESIS

Department of Civil Engineering

Civil Engineering (English) Program

Supervisor

Assoc. Prof. Dr. Zeynep IŞIK

February, 2022

REPUBLIC OF TURKEY

YILDIZ TECHNICAL UNIVERSITY

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

DEVELOPMENT OF A BIM BASED DELAY ANALYSIS METHOD IN CONSTRUCTION INDUSTRY

A thesis submitted by Murat ÇEVİKBAŞ in partial fulfillment of the requirements for the degree of DOCTOR OF PHILOSOPHY is approved by the committee on 02.02.2022 in Department of Civil Engineering, Civil Engineering (English) Program.

Assoc. Prof. Dr. Zeynep IŞIK
Yıldız Technical University
Supervisor

Approved By the Examining Committee

Assoc. Prof. Dr. Zeynep IŞIK, Supervisor
Yıldız Technical University ____________________

Assist. Dr. Hande ALADAĞ, Member
Yıldız Technical University ____________________

Prof. Dr. Hüsnü Murat GÜNAYDIN, Member
Istanbul Technical University ____________________

Prof. Dr. Gül Polat TATAR, Member
Istanbul Technical University ____________________

Assoc. Prof. Dr. Almula Köksal IŞIKKAYA, Member
Yıldız Technical University ____________________

I hereby declare that I have obtained the required legal permissions during data collection and exploitation procedures, that I have made the in-text citations and cited the references properly, that I haven't falsified and/or fabricated research data and results of the study and that I have abided by the principles of the scientific research and ethics during my Thesis Study under the title of A Dynamic Delay Analysis Selection Method in Construction supervised by my supervisor, Assoc. Prof. Dr. Zeynep IŞIK. In the case of a discovery of false statement, I am to acknowledge any legal consequence.

Murat ÇEVİKBAŞ

This study was supported by The Coordinatorship of Scientific Research Projects of YTU (FDK-2021-4336).

ACKNOWLEDGEMENTS

It is a pleasure to express my acknowledgement for those who made this study possible.

First and foremost, I would like to express my profound gratitude and sincere appreciation to my thesis supervisor Assoc. Prof. Dr. Zeynep IŞIK for guiding and inspiring me to accomplish this study through her profound knowledge along with her valuable feedbacks and continuous constructive supports.

I also owe a great debt of gratitude to my wife Lecturer Hacer ÇEVİKBAŞ and my son Yunus Kadir ÇEVİKBAŞ who gave me moral supports.

Murat ÇEVİKBAŞ

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

ECD	Excusable Compensable Delay
END	Excusable Non-compensable Delay
$MC_{(i)}$	Commencement Date of $Activity_{(i)}$ in Modified Schedule
$MC_{D(i+1)}$	Commencement Date of Driven Successor Activity of $Activity_{(i)}$ in Modified Schedule
$MF_{(i)}$	Finish Date of $Activity_{(i)}$ in Modified Schedule
$MF_{D(i-1)}$	Finish Date of Driven Successor Activity of $Activity_{(i)}$ in Modified Schedule
MS	Modified Schedule
MSvsMUS	Modified Schedule Versus Modified Updated Schedule
MUS	Modified Updated Schedule
NED	Non-excusable Delay
$OD_{(i)}$	Original Duration of $Activity_{(i)}$ in Baseline Schedule
$SD_{(i)}$	Subnet Duration of $Activity_{(i)}$ in Subnet Schedule
TIA	Time Impact Analysis
$UF_{(i)}$	Finish Date of $Activity_{(i)}$ in Modified Updated Schedule
$UF_{(i-1)}$	Finish Date of Driven Predecessor of $Activity_{(i)}$ in Modified Updated Schedule
$US_{(i)}$	Start Date of $Activity_{(i)}$ in Modified Updated Schedule
$US_{(i-1)}$	Finish Date of Driven Predecessor of $Activity_{(i)}$ in Modified Updated Schedule
WA	Windows Analysis
WCD	Windows Commencement Date
WFD	Windows Finish Date
$\alpha F_{(1i)}$	Finish Date of ECDs and ENDs in Modified Schedule
$\alpha F_{(2i)}$	Finish Date of ECDs and ENDs in Modified Updated Schedule
$\alpha S_{(1i)}$	Start Date of ECDs and ENDs in Modified Schedule
$\alpha S_{(2i)}$	Start Date of ECDs and ENDs in Modified Updated Schedule
$\alpha F_{2D(i-1)}$	Finish Date of Driven Predecessor Activity of ECDs and ENDs in Modified Updated Schedule
$\alpha S_{2D(i-1)}$	Start Date of Driven Predecessor Activity of ECDs and ENDs in Modified Updated Schedule

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Development of a BIM Based Delay Analysis Method in Construction Industry

Murat ÇEVİKBAŞ

Department of Civil Engineering

Doctor of Philosophy Thesis

Supervisor: Assoc. Prof. Dr. Zeynep IŞIK

In this thesis, as part of information gathering concerning delay analyses in the construction industry, the current problems, gaps and trends were detected under two researches, which enabled to set the core objective of this study. Firstly, Scientometric Analysis was conducted via VOSviewer software. It was concluded that there are gaps and trends in literature towards developing new delay analyses because of the challenges and drawbacks of the existing delay analysis methods. Secondly, an expert review meeting with 7 experts was conducted and common themes of the participants' views are presented in this study to detect further problems concerning delay analysis methods. All in all, in this study, the problems related to current delay analyses were highlighted. It is clear that it is difficult to conduct reliable analyses without overcoming the detected problems. Therefore, in this study, it is aimed to develop a new delay analysis method named Modified Schedule vs Modified Updated Schedule (MSvsMUS), which overcomes the detected drawbacks to be able to compute the project delays and apportion the project delays to the contracting parties systematically and efficiently. Following the development of the method, the method was validated through a hypothetical study, a real case study and an expert panel. On the whole, MSvsMUS was regarded as applicable by all of the participants. Afterwards, the developed method was integrated into 4D-BIM software, namely Bexel Manager by using the programing language of ‘C++’. The integration of the method into 4D-BIM software was successfully tested with a real dwelling project as a case study. Developing a new delay analysis method and integrating it into BIM software, this study is believed to provide a significant improvement in the construction industry by presenting solutions for detected problems faced in the current delay analysis applications.

Keywords: BIM, Delay Analysis, Delay, Dispute, Construction

YILDIZ TECHNICAL UNIVERSITY

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

ÖZET

İnşaat Sektöründe BIM Tabanlı Gecikme Analizi Yönteminin Geliştirilmesi

Murat ÇEVİKBAŞ

İnşaat Mühendisliği Bölümü

Doktora Tezi

Danışman: Doç. Dr. Zeynep IŞIK

Bu tezde, gecikme analizleri ile ilgili inşaat sektöründeki mevcut sorunlar, literatürdeki boşluklar ve eğilimler bilgi toplama kapsamında iki araştırma altında tespit edildi ve bu, çalışmanın temel amacını belirledi. İlk olarak VOSviewer yazılımı ile Scientometric Analizi yapılmıştır. Mevcut gecikme analizi yöntemlerinin zorlukları ve sakıncaları nedeniyle literatürde yeni gecikme analizleri geliştirmeye yönelik boşluklar ve eğilimler olduğu sonucuna varılmıştır. İkinci olarak, gecikme analizi yöntemleri ile ilgili daha fazla sorunu tespit etmek için 7 uzmanla birlikte bir uzman inceleme toplantısı yapılmış ve katılımcıların görüşlerindeki ortak temalar bu çalışmada sunulmuştur. Sonuç olarak, bu çalışmada mevcut gecikme analizleri ile ilgili sorunlar vurgulanmıştır. Tespit edilen problemlerin üstesinden gelmeden güvenilir analizler yapmanın zor olduğu açıktır. Bu nedenle, bu çalışmada, proje gecikmelerini hesaplayabilmek ve proje gecikmelerini sözleşme taraflarına sistematik ve verimli bir şekilde bölüştürebilmek için, tespit edilen eksiklikleri gideren, Modified Schedule vs Modified Updated Schedule (MSvsMUS) adlı yeni bir gecikme analizi yönteminin geliştirilmesi amaçlanmıştır. Metodun geliştirilmesinin ardından, varsayımsal bir çalışma, gerçek bir vaka çalışması ve bir uzman paneli aracılığıyla yöntemin geçerliği sağlanmıştır. Genel olarak, MSvsMUS tüm katılımcılar tarafından uygulanabilir olarak kabul edildi. Daha sonra geliştirilen yöntem, "C++" programlama dili kullanılarak Bexel Manager adlı 4D-BIM yazılımına entegre edilmiştir. Geliştirilen yöntemin 4D-BIM yazılımına entegrasyonu vaka çalışması olarak gerçek bir konut projesi ile başarıyla test edilmiştir. Yeni bir gecikme analiz yöntemi geliştiren ve bu modeli BIM yazılımına entegre eden bu çalışmanın, mevcut gecikme analizi uygulamalarında karşılaşılan tespit edilen sorunlara çözüm sunarak inşaat sektöründe önemli bir gelişme sağlayacağına inanılmaktadır.

Anahtar Kelimeler: BIM, Gecikme Analizi, İhtilaf, İnşaat

1
INTRODUCTION

One of the main goals of project management is to reach the project objectives by using limited resources effectively and efficiently. Managing the resources in construction projects containing high multi-stakeholder interactions requires an extremely complex management process and may bring uncertainties in projects. Consequently, changes occurring during the course of the projects are inevitable [1]. These changes mostly end up with conflict(s) and claim(s) between the parties. Delay is considered to be one of the major causes of the claims and also one of the most challenging issues to overcome in the construction industry [2]–[5]. Researchers have made tremendous efforts for developing delay analysis techniques and documents of good practice in order to guide practitioners instrumentally. However, the resolution of the delay is still one of the most difficult issues to overcome in construction projects [2], [3]. One of the most striking reasons for these difficulties is the constant change on the critical path of the project work schedule and the effect of the delayed activities on the completion date of the projects [6]. Especially, in big-sized construction projects, elaborating the facts of the delays and apportioning the delays to the responsible parties are very difficult and time-consuming processes [7]. It is vital to determine the amount of the delay along with its contractual responsibility in order to reduce the disagreements in the projects and to execute the project as per the expectations of the stakeholders [8]. When the delay is not shared out to the responsible parties thoroughly, parties tend to seek their rights in the courthouse which is a very time and cost-consuming process [9]. Since delay analysis is one of the most challenging and vital topics in the construction industry, researches in this matter are very valuable to maintain the fragile structure of the construction industry.

1.1 Literature Review

Involving sophisticated multidisciplinary works, construction projects are likely to contain uncertainties which make them prone to delays [9], [10]. Numerous studies have been conducted by researchers on the delay topic which is one of the primary areas of scientific studies due to the potential undesirable and devastating effects of delays on time and cost in construction projects [11]–[13]. On the one hand, techniques and effectiveness of the existing delay analysis methods have been investigated and widely discussed in the literature [14]–[23] and new delay analysis methods have been developed to overcome the flaws of the former delay analysis methods and thus surge the efficiency in practice [4], [7], [24]–[34]. Furthermore, studies contributing to the selection of the most appropriate delay analysis method have been conducted [14], [35]–[38]. But on the other hand, the occurring disadvantages of the current delay analysis methods have made the project contracting parties still suffer from computing and apportioning the delays [20], [39]. In other words, despite all these studies, the occurring disadvantages of the current delay analysis methods couldn't help prevent the disagreements concerning computing the delays and apportioning the delays between the contracting parties [20], [39]. In order to settle these disagreements, it is highlighted in the literature that a delay analysis method should take attributes such as consideration of Critical Path (CP) [40]–[42], actual progress data [43]–[45] and fluctuations on CP [28], [40], [46] into account in order to offer reliable outcomes. Additionally, due to lack of the procedure, delay analysis methods may yield distinct results even if the same delay analysis method is employed [11] and this is likely to result in inconsistency in construction projects [21], [47]. Due to the inadequacies of delay analyses disregarding the defined attributes, accuracies of the existing delay analysis methods have been questioned by the practitioners and courts [40], [48].

Delay analysis is mostly used with scheduling software. One of the current trends concerning scheduling tasks is 4D-BIM software. BIM has gained widespread attention because of the UK governments' mandate to use BIM on public sector projects since 2016 [49]. Tarek [50] conducted delay analysis with BIM by adopting Impacted As-Planned delay analysis method in his doctorate thesis. Concerning BIM program, Revit 2017 was used. Delay analysis code was written by C programming language and inserted as an add-in to the Revit. Additionally, Gibbs [51] suggested that BIM software may be used to overcome the major challenges faced by a delay analyst, which are information gathering and representation of the analysis. Vacanas et al. [52] proposed the accommodating BIM software and Unmanned Aerial Vehicle (UAV) technologies in infrastructure projects during conducting the delay and disruption analysis. Chou and Yang [53] discussed the possibility of using BIM for delay analyses. To sum up, more data included in the delay analysis can often produce more accurate results. With the help of BIM offering a data-intensive visualization, it is believed that BIM is an ideal means to solve the problems encountered in the schedule delay analysis domain. Anderson [54] elaborated the use of 4D models in support of delay analyses and asserted that BIM overcomes communication problems, which is one of the major obstacles for delay analyses in the construction industry, through its visualizing attributes. In addition to this, 4D-BIM model was also detected to be effective to ease dispute resolution processes. Moreover, Marzouk [55] proposed a methodology to assist project parties in handling such claims in a proactive manner by combining a responsibility matrix of claims causes with 5D BIM model in order to minimize the impact of claims in construction projects. This model visualizes and foresees claims or potential claims.

One of the earliest researches concerning delay analysis is 'Management, Scheduling and Delay Claims', conducted by Leary Christopher P. in 1982. Between 1982 and 2020 (15th November), 160 documents published in total related to delay analyses in the construction industry are illustrated in Figure 1.1 below.

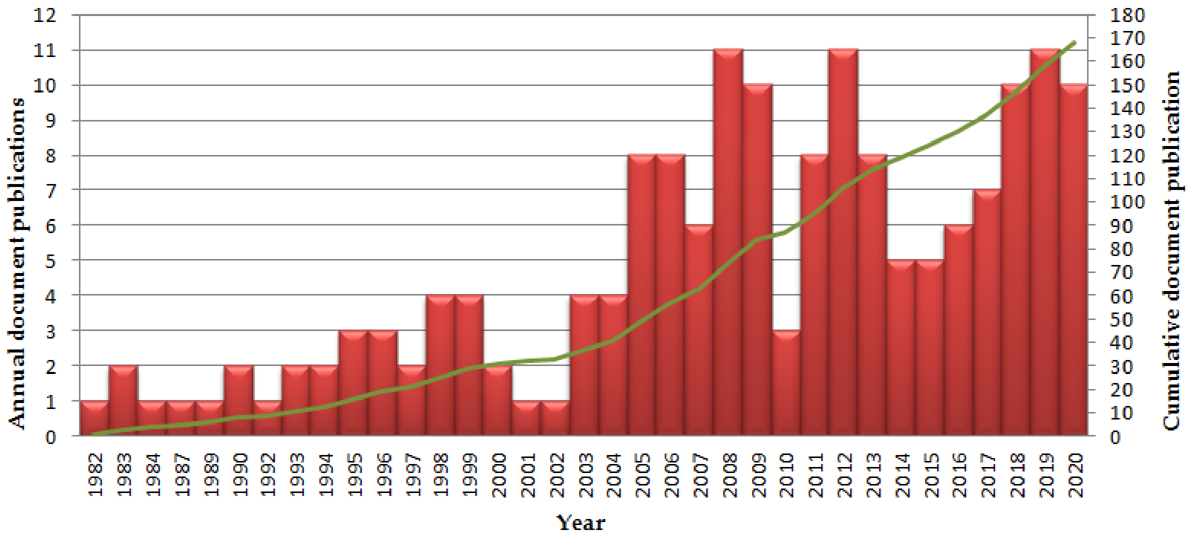


Figure 1.1 Distribution of the indexed researches published between the 20th and 21st Centuries

While the numbers related to the cumulative line illustrated in green colour are represented on the right side, the amounts of annually published documents are shown on the left side of the chart. It is explicitly inferred from the chart that the rate of publication concerning delay analyses in the construction industry has had an increasing trend over the years. Publications were observed to skyrocket after 2005. While the average of annual publications concerning delay analyses in the construction industry was 2.2 until 2005, the average of yearly publications surged to almost 8 between 2005 and 2020. Compared to the studies conducted in the 20th century, many more publications concerning delay analysis in the construction industry have been conducted in the 21st century. This is in line with the argument that delay analysis topic has become an increasingly needed research field, and growth in the number of publications is likely to continue in order to meet the needs in the industry and academia. A scientometric analysis is needed in order to gain insight into state-of-the-art developments by processing very large amounts of heterogeneous data which has been growing rapidly; thus, this enables the researchers and practitioners to get better decisions on delay analysis.

1.2 Objectives of the Thesis

Being unique in nature, construction projects involve high risks affecting the completion date of the projects. 98% of mega construction projects were delayed or over budget [60]. Although numerous studies concerning the delay domain - which is one of the most frequent dispute areas in the construction industry – have been conducted, the construction industry still suffers from the inadequacies resulting from delays. One of the mediums concerning settling the delay-related dispute is delay analysis. Since numerous studies have been conducted to improve delay analysis topic in the construction industry, researchers and practitioners still highlight the drawbacks of the existing delay analysis methods as investigated in Section 1.2. Moreover, delay analysis is dependent on scheduling software; however, 4D-BIM – which allows managing scheduling tasks - has a rising trend to be used in construction projects. Therefore, this study has a research question: Can a BIM-based delay analysis model which overcomes the defined problems be developed? Based on this question, for resolving the delays in construction projects in a proactive manner, it is aimed to develop a new and BIM-based delay analysis method named as “Modified Schedule versus Modified Updated Schedule (MSvsMUS)” in this thesis. This objective of the thesis is categorized under two main headings as followings.

Objective 1 – A new and more comprehensive delay analysis method - which eliminates the aforementioned defects of the other delay analyses – is aimed to develop along with a procedure and numerical model.

Objective 2 – The developed delay analysis method is aimed to be integrated into BIM software in order to obtain a BIM-based delay analysis model.

1.3 Hypothesis

Change - which means any addition, omission and modification to a contract - may affect the scope of the contract and this often ends up with an adjustment to the price and completion date of the projects [61]. Delays are the major causes of claims and disputes in construction projects [36]. The resolution of time and cost related claims resulting from changes in the projects remains a difficult commitment for all project parties [62]. This may result in friction and even litigation between the contracting parties. Many different delay analysis methods are available and selecting the suitable one is still the concern of contracting parties due to the strengths and weaknesses of the existing delay analysis methods [35]. Drawbacks of the existing delay analysis methods were highlighted in Section 1.2, and it is very certain that aforementioned drawbacks make the construction projects suffer from delays [48]. By considering the abovementioned objective of this study, the hypotheses put forward in this context are as follows:

1. The existing delay analysis methods are far from practical use for settling the delay-related dispute due to their current disadvantages. Considering the relationships among the activities, a new delay analysis model being capable of computing the variations between the planned fragnet/s and actualized fragnet/s periodically is able to overcome the detected problems of the current delay analysis methods

including one of the major drawbacks which refers to lack of computing any further delay or improvement made by the contractor for the fragnet/s.

2. The existing solutions concerning the delay analysis can be improved with the help of Delay Analysis conducted with 4D-BIM. The numerical model and algorithm of the new delay analysis model is foreseen to be transferred into BIM software as an add-in tool by using programing languages.

While Hypothesis 1 corresponds to Objective 1, Hypothesis 2 corresponds to Objective 2.

1.4 Problem Justification

Notwithstanding the undeniable contribution of the construction industry to an economy, it is highly probable to prone to disputes which may conclude with cost and time overrun due to the complicated processes of the construction projects [56], [57]. Delay is the main source of dispute and its proactive and timely resolution plays a core role in business success. However, most projects fail in resolving the time-related disputes [58] and this interminable issue in construction projects triggers the researchers' attention to concentrate on the studies to overcome time-related disputes in construction projects. As is seen in literature, there has been steady and growing interest in the researches tackling the delay analysis contexts with different aspects. While some researchers have focused their attention on detecting the reasons for time-related disputes in order to prevent disputes during the course of the projects, other researchers have concentrated on detecting strengths and weaknesses of the current delay analysis models, developing more robust models than the existing models and identifying the most appropriate delay analysis methods for construction projects. Although such attentions are desirable for the sake of the construction industry, the accumulation of the volume of researches in this field results in many challenges. Due to the versatility of the studies, an overarching study is required to detect the major drawbacks concerning delay analysis domain.

Detecting and overcoming the drawbacks of the current delay analysis models are one of the most up to date trends concerning delay analysis. Although there are numerous studies over delay analysis, the existing drawbacks concerning delay analyses complicate the resolution of disputes arising from delays in projects. Therefore, the construction projects mostly fail to apportion the project delays to the project parties due to the disadvantages of current delay analyses and their limited recognition [15], [17].

In this chapter, an in-depth literature review was conducted to figure out the major drawbacks of the existing delay analyses. In summary, it is inferred from the literature that none of the current delay analysis models can detect any improvements or further delays made by the contractor for the subnets. Also, ignoring attributes such as CPM, the actual sequence of work, fluctuations on Critical Path (CP) and concurrent delay and float ownership were also detected as flaws for some of the existing delay analyses. These also reduce the reliability of delay analysis models. Additionally, one of the detected drawbacks is the lack of delay analysis implementation in BIM domain. Only Impacted As Planned delay analysis method was integrated into BIM software. Although many projects conduct the scheduling tasks via 4D-BIM, 4D-BIM tools are not capable of implementing the adopted delay analysis models such as Time Impact and Windows Analysis. Later, all delay analysis methods are illustrated via Gantt Charts and/or set of equations without considering the all types of relationships of the activities such as Finish to Start (FS), Start to Start (SS) and Finish to Finish (FF) and this reduces the reliability of delay analysis models. In the wake of the in-depth literature review, the following drawbacks were also detected. Firstly, some of the delay analysis models consider only the owner's delays, and this result in dispute between the parties due to overlooking the concurrency. Secondly, most of the delay analysis methods don't have any procedure to ensure their consistent practical applications. Lacking the procedure and numerical explanation of a delay analysis model result in excessive requirements which are time, cost and manpower of experts. It should be noted that the validity of the aforementioned drawbacks will be further verified in the chapters concerning scientometric analysis and unstructured interviews conducted with the construction practitioners.

By looking at the trend detected in this thesis, a new delay analysis model which has procedures and algorithms and solves the drawbacks of the current delay analyses seems to be needed. In addition to the application of delay analysis models with scheduling software, the usage of delay analysis methods in construction projects with 4D-BIM which includes scheduling tasks has now come to the forefront. BIM has gained widespread attention by not only practitioners but also governments. For instance, the UK government has been forcing to use BIM in the public sector projects since 2016 [49]. Delay analysis still depends on scheduling software today; however, usage of scheduling software has been diminishing day by day because there is an upsurge need concerning using 4D-BIM for scheduling issues in construction projects. Applications of BIM reduce the probability of dispute occurrence, as it offers detection of conflict and early error [59]. Additionally, BIM overcomes the problems concerning record keeping, data analysis and data presentation encountered in the application of traditional delay analysis methods

; therefore, BIM is a candidate to be an ideal tool to facilitate schedule delay analysis [53]. Although BIM, which includes scheduling supports with 4D modeling and overcomes the problems resulting from scheduling software, is widely used in the construction industry, there is no study showing how to use the most accepted delay analysis models like Windows Analysis and Time Impact Analysis with BIM. On the whole, a delay analysis model conducted without considering these issues will not give realistic and satisfactory results.

1.5 Research Methodology

In order to justify the existing drawbacks of the delay analysis methods provided in Section 1.2, this study adopted two methodologies. Firstly, a scientometric analysis was conducted to detect the current problems as well as gaps and trends concerning the delay analysis domain. Secondly, unstructured interviews were made with 7 experts to further detect the drawbacks of the current delay analysis methods. Afterwards, resolving detected drawbacks, a new delay analysis method was developed numerically along with procedure and pseudo-codes. Later, the developed method was validated with a hypothetical study by comparing the outcomes of the developed method (MSvsMUS) with Windows Delay Analysis which is one of the most adopted delay analysis methods by courts [44]. Next, the developed method was presented through a case study which is a real airport project. Afterwards, an expert panel was conducted with 7 experts to evaluate the effectiveness and applicability of the developed method by elaborating the outcomes of the case study. Furthermore, the developed delay analysis method was integrated into a 4D-BIM software, namely Bexcel Manager, as an add-in tool by using C++ programming language. Lastly, the add-in tool was validated via a real case study which is a dwelling project and it was concluded that the developed method and add-in tool of 4D-BIM were successfully validated.

1.6 Limitation of the Research

The limitation of this study is mainly based on the data collection process while detecting the drawbacks of the delay analysis methods. Two studies were conducted, which are scientometric analysis and unstructured interviews with 7 experts. When the number of experts interviewed increases, the number of detected drawbacks can upsurge and this enables to develop a more comprehensive delay analysis method. In addition to this, an add-in tool was developed for only Bexel Manager; therefore, new add-in tools can be developed for the other 4D-BIM software by using the provided codes in this thesis.

1.7 Organization of the Thesis

This thesis consists of 9 chapters. In Chapter 1, introductory information is provided to summarise the thesis with justification of the problem, objectives and hypothesis based on the literature review. Methodology, limitations of the research and organization of the thesis are also provided in this chapter. Additionally, in order to enable the reader to further understand the outputs of this study, the existing delay analysis methods are defined, formulized and categorized in Chapter 2, and BIM domain is investigated in Chapter 3 respectively. Aftrewards, In Chapter 4, the methodology of this study – which includes scientometric review, unstructured interviews, development of a new delay analysis method, namely MSvsMUS, and integration of it into BIM software as an add-in tool – is presented. Next, In Chapter 5, the findings of this study as per the headings of the methodology section are demonstrated. Later, in Chapter 6, while the developed method is validated with a hypothetical study, case study and an expert panel, the developed add-in tool for 4D-BIM is validated with a case study. Moreover, the conclusion and recommendation section of this study is provided in Chapter 7. Furthermore, pseudo-codes of the developed method are provided in Appendix, namely "Developing Pseudo Code" to enable the reader to develop further software concerning the developed method. Lastly, codes of the developed method in 4D-BIM are presented to enable the reader to improve the software concerning 4D-BIM software used in this thesis in Appendix, namely "Developing Core Functionality".

2
EXISTING DELAY ANALYSIS METHODS

Claim is an inevitable part of construction projects due to its complex nature. As for the delay in the construction projects, it is considered to be one of the major causes of claim and also one of the most challenging issues to overcome in construction industry [2], [3]. For a delay to be settled and equitably segregated as per project parties, the most appropriate delay analysis method considering the characteristics of the project environment is to be adopted. Researchers have made tremendous efforts for developing delay analysis techniques and documents of good practice in order to guide practitioners instrumentally. Despite the availability of these delay analyses, the projects mostly fail to apportion the project delays to the responsibilities of the project parties due to the disadvantages of current delay analyses and their limited recognition [15]. According to a survey conducted with construction consulting organizations in the UK, most of the practitioners have lack of recognition of delay analysis methods despite their importance [17]. The fact that the projects suffer from the catastrophic results of the delays has attracted the attention of the researchers and paved the way for the improvement of delay analysis concerning construction projects.

Determining the delay types of the delaying events is a challenging task because of the associated problems with apportioning the responsibility of delays to the parties [8]. Delays are categorized as per liability into two major types which are excusable or non-excusable delays [63]–[65] as is illustrated in the figure below.

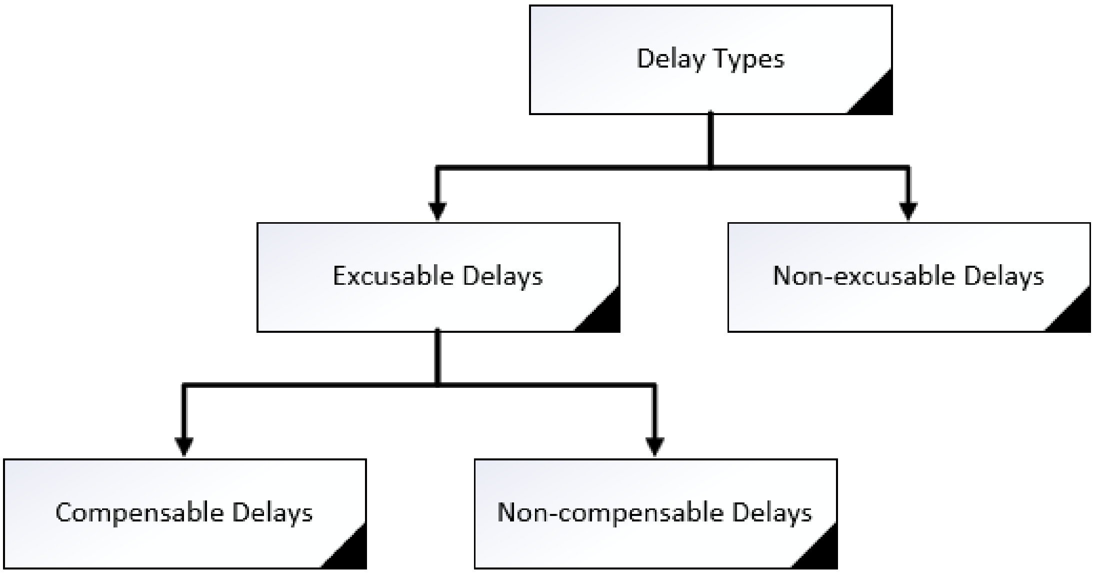


Figure 2.1 Delay types

Excusable delay events are beyond the control of a contractor. When an excusable delay occurs and affects the completion date of a project, the contractor is entitled to an extension of time. Excusable delays can be further categorized as compensable and non-compensable delays [9], [66], [67]. While A Non-excusable delay (NED), which is also known as Culpable Delay [44], results from contractor's actions, an ECD and an END are caused by the owner and Acts of God respectively [68]. While non-excusable delays may arise from slow progress, compensatory delays may result from change order, site instruction, disruptions and suspensions; however, EDs may result from the events which are outside the control of contractors and clients, such as strikes, fire, differing site conditions and change in government law, etc. Particular construction contract documents need to be checked to identify EDs in a project. In the case of an ED, the contractor is entitled to EOT. Few amounts of delay analyses have been developed to figure out EDs and CDs and NDs to apportion the delays to the clients and contractors. Attributes of the existing delay analysis methods are given below.

2.1 Global Impact

The Global Impact Method is one of the simple ways to calculate the Extension of Time (EOT) in a project. This method is not Critical Path Method (CPM) grounded. Summing the durations of delayed activities provide EOT for a project [9] without considering the concurrent delays in this method. This method has some major drawbacks as follows; It may exaggerate the overall delays as it disregards the critical path of the schedule, concurrent delays, actual site progress and contractor's delays [19]. This method is not considered reliable for the construction projects containing a large amount of delaying events and mostly it is resisted by courts and arbitration panels [35].

2.2 Net Impact

Net Impact Method basically has the same characteristics as the Global Impact Method. The only difference is that in the case of concurrent delays, only the delay having the maximum duration is taken into consideration [18]. Considering the concurrent delays is one of the advantages of this method over Global Impact Method.

2.3 Impacted As-Planned (IAP)

IAP Method utilizes the baseline schedule and project records in terms of time. This method enables project managers to compare the new completion date derived from this method with the original baseline completion date as per client-owned and contractor-owned delays [6], [47]. Vasilyeva-lyulina et al. [47] identify the critical mechanism of IAP by developing a basic equation to avoid the ambiguities in implementations in practice.

In order to compute EOT, compensatory delays and excusable delays (if the contract allows) are incorporated into the baseline schedule. In order to determine the delays from the owner's perspective, non-excusable delays and mitigations made by the contractors are to be injected into the baseline schedule; however, finding the contractors' delays doesn't seem that much practical, since it requires including all actual durations of the activities in the project schedule. The courts in the US widely discard its use [69], [70].

2.4 As-Planned vs As-Built

As-Planned vs As-Built method is also known as Total Time Approach, Adjusted As-Built CPM Technique and Impacted As-Built Technique. This method is CPM grounded and requires delaying events to be incorporated into As-Built Schedule. This method has a simple calculation and compares As-Built Schedule and Baseline Schedule to find the delays; however, the value of total delays is not able to be directly apportioned to the contractors and clients without an in-depth elaboration. The activities on Critical Path (CP) in Baseline Schedule and As-Built Schedule are compared in order to compute the total impact of delays on the project and allocate the delays to the owner and contractor [18].

2.5 Windows Analysis

Windows Analysis, which is known as Time Slice Analysis, Snapshot Technique and Contemporaneous Analysis Method, is a very robust method in terms of detailing the delays as per responsible parties. It is a CPM based method, and as-built events, client delays, contractor delays and excusable delays are incorporated into Baseline Schedule [31] in order to be compared with the Baseline Schedule for each window. It divides the total project duration into digestible time periods (windows or snapshot) in order to examine the delays [34], [40]. Delays encountered in each window are computed [20] in order to segregate the delays as per client and contractor. Mostly, the window size is adjusted according to milestones, schedule updates or major delay events, and the outcome derived from each window constitutes the overall project delays [34].

2.6 Modified Windows Schedule Analysis

Gothand [19] developed Modified Windows Schedule Analysis which includes delaying events of both claimants and defendants in order to make the analysis more reliable from the point of defendants. The analytical process of Modified Windows Schedule contains the following steps which are; the establishment of baseline schedule incorporated with the windows based fragnets of delaying events to produce Impacted Schedule, updating the Impacted Schedule with actual site progress, comparing the Baseline Schedule and updated Impacted Schedule to find delays as per owner and contractor and compute contractor's loss or gain for each window, and summing up all the delays and improvements for all windows.

2.7 Daily Windows Delay Analysis

Hegazy and Zhang [34] and Hegazy et al. [31] proposed Daily Windows Delay Analysis which is based on Intelligent Bar Chart Method involving daily progress, daily delays, responsible parties of delays and any other related data. The main objective of this method is to set the daily window span. This method requires daily record keeping and daily action to conduct in contrast to the other methods; thus, it is more precise since it is not affected by the fluctuations in the critical path like may be seen in Windows Analysis [34].

2.8 Daily Windows Analysis with Multiple Baseline Updates

Hegazy and Menesi [28] developed Daily 'Windows Analysis with Multiple Baseline Updates' which takes multiple baselines and resource allocations into account in order to consider every change in relations and durations of the schedule activities. The systematic procedure of this method contains the continuous daily baseline production each of which involves the addition of logical relationship and activity duration as per site progress over the previous baseline. Then, Daily Windows Analysis is applied on a daily basis to compare with the new baseline. Numerical model of this method was developed and introduced into Microsoft Excel with macros by Hegazy [71]; however, only Finish-to-Start (FS) relation, maximum 200 days of project duration and maximum 3 predecessor and 3 successor activities for each activity are allowed in the software, which is away from the practicality.

2.9 Time Impact Analysis (TIA)

The main idea for this method is to get a general overview concerning project completion date affected by major delaying events. TIA determines the impact of each delay right after its occurrence and requires the addition of each delaying event into the most recently updated schedule to be compared with the baseline schedule [72]. A numerical model of TIA was provided by Vasilyeva-lyulina et al. [47] to identify the basic concept of the critical mechanism. Although TIA has a very similar technique to Windows Analysis, the major distinction between them is that while the latter is a retrospective analysis which concentrates on what actually happened in the project, the former is for mainly prospective analysis which focuses on what might happen in terms of delay [15].

2.10 Collapse Analysis

Collapse Analysis, which is known as But-For Analysis, As-Built But For Analysis, As-Built Collapsing Technique and As-Built Subtracting Impacts, Collapsed As-Built [23], [47], [73], is based on an as-built schedule. Regarding the analytical procedure of this method, delays are removed from the as-built schedule to collapse the schedule, and the obtained schedule is compared with the baseline schedule in order to compute the project delays. Since As-Built Schedule is utilized, this method is only conducted for retrospective analysis [74]. With respect to delays from the contractor's point of view, non-excusable delays are to be removed [37]. On the contrary, to compute the delays from the owner's point of view, excusable delays and compensatory delays are removed from the as-built schedule [30]. Vasilyeva-lyulina et al. [47] developed a basic numerical model of Collapse Analysis to maintain its standard implementation in construction industry.

2.11 Modified But-For (MBF)

Mbabazi et al. [26] created Modified But-For (MBF) analysis. In addition to traditional But-For method, it improves the areas concerning "Representation of Activity Disruption" and "Representation of Concurrent Critical Delays". This method takes the advantage of Venn diagram in order to represent the delays of parties namely, owner (o), contractor (c), neither party (n) and their combinations. When they are represented in a Venn diagram, totally 3 sets and 4 unions of sets are produced as following combinations; 'o', 'c', 'n', 'o+c', 'o+n', 'c+n', and 'o+c+n'. As in the process of But-For analysis, in order to compute the delays made by a party, other parties' delays are removed from the As-Built Schedule.

2.12 Isolated Delay Type (IDT)

Alkass et al. [9] developed Isolated Delay Type (IDT) method. IDT method intents to benefit from the techniques concerning the classification of delay types, concurrent delays and real-time CPM analysis. Like Collapse But-For (CBF) method, IDT analyses the delays as per the owner's perspective (non-excusable delays) and contractor's perspective (excusable delays and compensable delays). This method takes the advantage of actual project records by injecting them into the Baseline Schedule. This modified schedule is an adjusted as-planned schedule and it is the basis for delay comparisons. To compute the delays from the client's perspective, non-excusable delays are incorporated into the baseline schedule to be compared with the original baseline schedule for each time period. The first IDT analysis is the base for the second IDT analysis. For the second period, any change in the logic or activity duration is to be reflected in the second period of IDT schedule. Additionally, since excusable delays are accepted by the parties, excusable delays which occur in the first period of IDT are to be injected into the schedule. With respect to the contractor's point of view, only EDs and CDs along with the duration and logic changes are reflected in the Baseline schedule for each time period.

2.13 Isolated Collapsed But-For (ICBF)

Yang and Yin [30] proposed Isolated Collapsed But-For (ICBF) method which uses the concept of IDT and Collapse Analysis; thus, this technique obtains the advantages which are associated with Collapse Analysis and IDT methods. Instead of utilizing a Baseline Schedule to be incorporated with client-based delays or contractor-based delays like IDT, As-Built schedule is simply used since it already has the actual data. In this method, project duration is divided into digestible periods. For each period, As-Built data is kept and the attributes of activities concerning latter periods are changed just like the attributes of activities in Baseline Schedule to obtain Adjusted Schedule. Then, either Excusable Delays or Non-Excusable Delays are incorporated to find the Adjusted Schedule with delays. At last, Adjusted Schedule and Adjusted Schedule with delays are compared to compute the delays of contractors and owners. Summing up all the periodical delays of contractors and owners provides the total delays of the concerned parties.

2.14 Delay Analysis Method Using Delay Section (DAMUDS)

Kim et al. [2] developed a method named as Delay Analysis Method Using Delay Section (DAMUDS). In order to consider concurrent delays and time-shortened activities in a more precise way, DAMUDS offers two new concepts concerning delay section (DS) and the contractor's float (CF) [40]. This method conducts the analysis separately for the activities without delay, the activities having a single delay, and the activities having the concurrent delays on DS. With respect to analysis of DAMUDS, baseline schedule is updated according to actual site progress. Updated activities without any delay in each DS provide information concerning either no delay or time-shortened durations which require to be added to CF. Regarding DS containing a single delay, if the period of DS is less than Total Float (TF) of the delayed activities (if total project duration remains the same after rescheduling), the same process with respect to activities without any delay is applied; otherwise, the delay is added to the responsible party. With regards to DS holding the concurrent delays of two or more activities, if total project duration is changed after rescheduling, delays are to be apportioned to the client and contractor. If TF>DS in all conditions, time-shortened durations, which are part of CF, are computed in each DS.

2.15 Accumulated Delay Analysis Method (ADAM)

Lee [27] offered Accumulated Delay Analysis Method (ADAM) using Linear Scheduling Method (LSM) to apportion delays considering the changes in productivity of each delayed activity to the concerned parties. At the outset, delay types are determined as per the lost productivity or pure interruption. Then their productivity functions are determined for both Baseline Schedule and As-Built Schedule in order to compute the accumulated delay.

2.16 Effect-Based Delay Analysis Method (EDAM)

EDAM method proposed by Yang and Kao [33] is a Windows-based delay analysis method. This method utilizes baseline schedule and delay attributes concerning start, finish and liability of delays. In order to compute the delay with its owned party, updated impacted baseline schedule and baseline schedule are compared for each window span. In this method, while windows containing no delay are analyzed for detecting the acceleration and mitigation, windows containing delays are analyzed on a daily basis to compute and apportion the delays as per responsible parties. A mathematical model and algorithm of EDAM are provided by Yang and Kao [33]; therefore, it is eligible to compute the delays via computers.

2.17 As-Planned But-For

This method requires delayed events (either CD and ED or ND) to be incorporated in the baseline schedule. The completion date of As-Planned But-For schedule is compared with the actual completion date [75]. In order to compute the contractor based delays, CD and ED are injected to the baseline schedule and completion date is deducted from the actual completion date. To calculate owner based delays, ND is incorporated to the baseline schedule and its completion date is subtracted from the actual completion date.

2.18 Stochastic Delay Analysis and Forecast Method (SDAF)

SDAF is a probabilistic method developed by Yang and Teng [29]. It combines Three Time Estimate Approach concerning Program Evaluation and Review Technique (PERT) for duration calculation and numerical delay analysis, using As-Planned schedule and As-Built schedule, invented by Shi et al. [76]. Compared to other methods, SDAF considers the possible variation of activity duration in order to improve the accuracy of assigning the delay liability. On the other hand, it is difficult to gather data concerning pessimistic, optimistic and most likely durations of activities in order to produce As-Planned Schedule in projects.

In addition to former delay analysis methods, Shi et al. [76] presented a method with respect to detecting delays in activities and evaluating their contributions to overall EOT in a project. The proposed method provides a set of equations which are not based on critical path analyses. This method compares the as-planned schedule with the as-built schedule of a project. As opposed to traditional delay analysis methods, it does not require a critical path and schedule update; instead, only finish-to-start logic is adopted in this method disregarding the other logics of start-to-start, finish-to-finish and start-to-finish.

With respect to analytical processes of delay analysis methods, Table 2.1 is given below to represent the basic concept of analytical process of each method along with their associated schedule techniques (CPM, LSM) and their outcomes of delays based on owners and contractors. Most of the techniques of delay analysis models are described in literature without any given numerical definitions, which results in ambiguities for their usage in practice. It is believed that basic numerical concept of each delay analysis provided in Table 2.1 will help maintain the standard and thus skyrocket the practicality of the techniques of these methods.

Table 2.1 An overview of the delay analysis methods

Method	Basic Concept	Models		Delay Owner		Schedule Technique				
						CPM			LSM	
		Static Models	Dynamic Models	Owner (CD and ED)	Contractor (ND)	Baseline Schedule	As-Built Schedule	Updated Schedule	Baseline Schedule	As-Built Schedule
Impacted As Planned (IAP)	Contractor's Point of View = Completion Date of Baseline Schedule – Completion Date of Baseline Schedule incorporated with EDs and CDs Owner's Point of View = Completion Date of Baseline Schedule – Completion Date of Baseline Schedule incorporated with NDs		✓	✓	✓	✓				
As Planned But-For	Contractor's Point of View = Actual Completion Date of Project – Completion Date of Impacted As Planned Schedule including NDs Owner's Point of View = Actual Completion Date of Project – Completion Date of Impacted As Planned Schedule including EDs and CDs		✓	✓	✓	✓				
Time Impact Analysis	Total Number of delays = n, $\sum_{i=1}^{n} (Delay_{\text{Baseline Schedule}-\text{Updated Schedule Including CDs and EDs}})_i$		✓	✓	✓			✓		

Table 2.1 An overview of the delay analysis methods (continued)

Method	Basic Concept	Models		Delay Owner		Schedule Technique				
						CPM			LSM	
		Static Models	Dynamic Models	Owner (CD and ED)	Contractor (ND)	Baseline Schedule	As-Built Schedule	Updated Schedule	Baseline Schedule	As-Built Schedule
Global Impact Analysis	Total Number of delays concerning EDs and CDs = n, $\sum_{n=1}^{n}(\text{Duration of Delay})_i$	✓		✓						
Net Impact Analysis	Total Number of delays concerning EDs and CDs = n, $\sum_{n=1}^{n}\left(\begin{matrix}\text{Durations of Delay taking the delay having the maximum} \\ \text{duration of concurrent delays into consideration}\end{matrix}\right)_i$	✓		✓						
As-Planned vs As-Built	As-Built Schedule including CDs, EDs and NDs - Baseline Schedule	✓		✓	✓	✓	✓			
Collapse Analysis	Owner's Point of View = Completion date of Baseline Schedule - As-Built Schedule removed CD and ED Contractor's Point of View = Completion date of Baseline Schedule - As-Built Schedule removed ND		✓	✓	✓	✓	✓			

Table 2.1 An overview of the delay analysis methods (continued)

Method	Basic Concept	Models		Delay Owner		Schedule Technique				
						CPM			LSM	
		Static Models	Dynamic Models	Owner (CD and ED)	Contractor (ND)	Baseline Schedule	As-Built Schedule	Updated Schedule	Baseline Schedule	As-Built Schedule
Windows Analysis	Total Number of window periods = n, $\sum_{i=1}^{n} (Delay_{\text{Baseline Schedule}-\text{Updated Schedule Including CDs,EDs and NDs}})\ i$		✓	✓	✓	✓		✓		
Isolated Delay Type (IDT)	Total Number of span as per either major delays or occurring series of delays = n, Owner's Point of View = $\sum_{i=1}^{n} (\text{Delay}_{\text{Baseline Schedule}-\text{ Baseline including NDs}})\ i$ Contractor's Point of View = $\sum_{i=1}^{n} (\text{Delay}_{\text{Baseline Schedule}-\text{ Baseline including CDs and EDs}})\ i$		✓	✓	✓	✓				

Table 2.1 An overview of the delay analysis methods (continued)

Method	Basic Concept	Models		Delay Owner		Schedule Technique				
						CPM			LSM	
		Static Models	Dynamic Models	Owner (CD and ED)	Contractor (ND)	Baseline Schedule	As-Built Schedule	Updated Schedule	Baseline Schedule	As-Built Schedule
Daily Windows Delay Analysis	Total Number of days in a Project= n, $\sum_{i=1}^{n}(\text{Delay}_{\text{Baseline Schedule}-\text{ Updated Schedule including CDs,EDs and NDs}})_{\text{i}}$		✓	✓	✓	✓		✓		
Windows Analysis with Multiple Baseline Updates	Total Number of days in the Project= n, $\sum_{i=1}^{n}(\text{New Baseline} - \text{Daily Windows Analysis concerning EDs, CDs and NDs})_{i}$		✓	✓	✓	✓		✓		
Modified Windows Schedule	Total Number of window periods = n, $\sum_{i=1}^{n}(\text{Delay}_{\text{Baseline Schedule}-\text{ Updated Baseline Schedule including fragnets of CDs, EDs and NDs}}$		✓	✓	✓	✓		✓		

Table 2.1 An overview of the delay analysis methods (continued)

Method	Basic Concept	Models		Delay Owner		Schedule Technique				
						CPM			LSM	
		Static Models	Dynamic Models	Owner (CD and ED)	Contractor (ND)	Baseline Schedule	As-Built Schedule	Updated Schedule	Baseline Schedule	As-Built Schedule
Delay Analysis Method Using Delay Section (DAMUDS)	Total Number of delay sections (DS) = n, $\sum_{i=1}^{n} (\text{Delay}_{\text{Baseline Schedule– Updated Schedule including CDs,EDs and NDs}})_i$		✓	✓	✓	✓		✓		
Modified But-For (MBF) Method	Baseline Schedule – As-Built Schedule (excluding the delays concerning owner, contractor, act of God or their combinations one at a time)		✓	✓	✓	✓	✓			
Accumulated Delay Analysis Method (ADAM)	$\sum_{n=1}^{n} (\text{Delay})_n - \frac{(\text{As Planned Productivity Function})_n}{\text{Production rate of AsPlanned Schedule}}$		✓	✓	✓				✓	✓

Table 2.1 An overview of the delay analysis methods (continued)

Method	Basic Concept	Models		Delay Owner		Schedule Technique				
						CPM			LSM	
		Static Models	Dynamic Models	Owner (CD and ED)	Contractor (ND)	Baseline Schedule	As-Built Schedule	Updated Schedule	Baseline Schedule	As-Built Schedule
Isolated Collapsed But-For (ICBF)	Total Number of digestible periods = n, Owner's Point of View = $\sum_{i=1}^{n} (\text{Delay}_{\text{Adjusted As–Built Schedule – Adjusted As–Built Schedule including NDs}})_i$ Contractor's Point of View = $\sum_{i=1}^{n} (\text{Delay}_{\text{Adjusted As–Built Schedule – Adjusted As–Built Schedule including CDs and EDs}})_i$		✓	✓	✓	✓	✓			
Effect-Based Delay Analysis Method (EDAM)	Total Number of windows = n, $\sum_{i=1}^{n} (\text{Delay}_{\text{Baseline Schedule– Updated Schedule including CDs,EDs and NDs}})_\text{i}$		✓	✓	✓	✓		✓		
SDAF Method	As-Built Schedule including CDs, EDs and NDs - Baseline Schedule (Durations are based on Three Time Estimate concerning PERT)		✓	✓	✓	✓	✓			

It should be noted that among the current delay analyses depicted in Table 2.1, selecting the most appropriate delay analysis method depends on the limitations of existing delay analyses presented in Table 2.3.

Following the in-depth literature review, the existing delay analysis models given in Table 2.2 are categorized according to the following criteria; type of analysis i.e. namely effect & cause or cause & effect cases, type of determination of the delay impact i.e. retrospectively or prospectively, type of determination of the critical path i.e. retrospectively or prospectively or contemporaneously, main delay analysis and their derivatives.

Table 2.2 Existing delay analysis methods and variations

Attributes of Delay Analyses				**Delay Analysis**	
Delay Analysis Number	**Type of analysis**	**Type of determination of the delay impact**	**Type of determination of the critical path**	**Main Types**	**Derivative**
	Effect & Cause				
		Retrospectively			
			Retrospectively		
1				Global Impact Analysis	
2					Net Impact Analysis
3				Collapse Analysis	
4					Modified But-For (MBF) method
5					As Planned But-For
			Contemporaneously		
6				As-Planned vs As-Built	
7					SDAF Method

Table 2.2 Existing delay analysis methods and variations (continued)

Attributes of Delay Analyses				**Delay Analysis**	
Delay Analysis Number	**Type of analysis**	**Type of determination of the delay impact**	**Type of determination of the critical path**	**Main Types**	**Derivative**
8				Windows Analysis	
9					Modified Windows Schedule
10					Daily Windows Delay Analysis
11					Total Float Management Technique
12					Windows Analysis with Multiple Baseline Updates
13					Delay Analysis Method Using Delay Section
14					Effect-Based Delay Analysis Method (EDAM)
15					Isolated Delay Type (IDT)
16					Isolated Collapsed But-For (ICBF)
17					Stochastic Delay Analysis and Forecast Method
18					Float ownership, logic change, and resource
19				Accumulated Delay Analysis Method (ADAM)	
	Cause & Effect				
		Prospectively			
			Prospectively		
20				Impacted As Planned (IAP)	
			Contemporaneously		
21				Time Impact Analysis (TIA)	

When the existing delay analysis methods are examined, it is detected that none of them can detect any improvements or further delays made by the contractor for the delays caused by the employer and force majeure. Delays calculated without making this determination will not give realistic results. When the mostly accepted delay analysis methods in construction industry are examined, the technique allowing the periodic elaboration of delays via delay analyses such as Windows Analysis and its derivatives is mostly adopted by the construction industry; however, conducting such manual techniques are time-consuming [34] and require numerical models and mostly a software development in order to be conducted in large and complex projects within a short period of time.

The drawbacks - which are associated with the existing delay analysis models – were detected via an in-depth literature review as depicted in Table 2.3.

Table 2.3 Major drawbacks of the existing delay analyses

Delay Analysis Methods	Does not compute any further delays or improvements made by the contractor for the fragnet/s	Ignoring the relationships of the activities	Available the 4D BIM Integration	Does not Analyze CPM	Ignores actual site progress	Ignores fluctuations on Critical Path	Does not involve a numerical explanation	Does not have an integrated software	Does not directly analyze the concurrent delays
Impacted As Planned	✓	✓	✓		✓	✓		✓	✓
As Planned But-For	✓	✓			✓	✓	✓	✓	✓
Time Impact Analysis	✓	✓						✓	✓

Table 2.3 Major drawbacks of the existing delay analyses (continued)

Delay Analysis Methods	Does not compute any further delays or improvements made by the contractor for the fragnet/s	Ignoring the relationships of the activities	Available the 4D BIM Integration	Does not Analyze CPM	Ignores actual site progress	Ignores fluctuations on Critical Path	Does not involve a numerical explanation	Does not have an integrated software	Does not directly analyze the concurrent delays
Global Impact Analysis	✓	✓		✓	✓	✓	✓	✓	✓
Net Impact Analysis	✓	✓		✓	✓	✓	✓	✓	✓
As-Planned vs As-Built	✓	✓				✓	✓	✓	
Collapse Analysis	✓	✓				✓		✓	✓
Windows Analysis	✓	✓					✓	✓	
Isolated Delay Type	✓	✓					✓	✓	✓
Daily Windows Delay Analysis	✓	✓					✓	✓	
Windows Analysis with Multiple Baseline Updates	✓	✓							

Table 2.3 Major drawbacks of the existing delay analyses (continued)

Delay Analysis Methods	Does not compute any further delays or improvements made by the contractor for the fragnet/s	Ignoring the relationships of the activities	Available the 4D BIM Integration	Does not Analyze CPM	Ignores actual site progress	Ignores fluctuations on Critical Path	Does not involve a numerical explanation	Does not have an integrated software	Does not directly analyze the concurrent delays
Modified Windows Schedule	✓	✓					✓	✓	
Delay Analysis Method Using Delay Section	✓	✓						✓	
Modified But-For method	✓	✓				✓			
Accumulated Delay Analysis Method	✓	✓		✓				✓	
Isolated Collapsed But-For	✓	✓						✓	
Effect-Based Delay Analysis Method	✓	✓						✓	
SDAF Method	✓	✓				✓		✓	

3
BUILDING INFORMATION MODELING (BIM)

BIM, which provides tools for professionals participating in the process concerning Architecture, Engineering and Construction (AEC) in order to facilitate design, construction and operation processes and form a reliable basis for decisions, is a three dimensional dynamic process modeling of building objects [77]. In other words, BIM is a process generating graphical and non-graphical information models in a Common Data Environment (CDE), which is a shared repository for digital project information and is formed in accordance with standards.

BIM usage is encouraged by a range of Public Policies to improve the efficiency of the construction sector and over the past 20 years, BIM implementation in the construction industry has become ubiquitous [78]. In order to further clarify the outputs of this thesis, BIM dimensions and 4D-BIM into which the developed delay analysis method is intended to be integrated are elaborated in this section.

3.1 BIM Dimensions

BIM Dimensions, which describe the features of BIM, consist of 3D, 4D, 5D, 6D and 7D models [79] [78].

The 3D model is a mock-up model articulating the design of the structure visually in three spatial dimensions, namely width, height and depth [78]. In other saying, BIM, which is a shared informational model, includes Z-axis to the existing X and Y axes to visualize the design in 3 dimensions. The construction industry consists of multidisciplinary works such as civil, electrical, mechanical, electromechanical and infrastructure. Projects combine these disciplines as per their nature from the design stage to the closing stage. During the life-cycle of projects, graphical (Level of Detail (LOD)) and non-graphical information (Level of Information (LOI)) are produced in the common data environment which is known as CDE via BIM technology in order to help the project stakeholders to manage multidisciplinary works more effectively [80]. Additionally, superimposing the design of different disciplines is a very time and cost-consuming issue in the projects and error is mostly inevitable during this overlaying process; however, BIM conducts clash detection in a very short time with a very detailed report. Moreover, during the closing phase of the projects, generated data via BIM is stored to manage future projects in a proactive manner.

The construction industry suffers from on-time delivery [81]. Schedule Management is one of the Project Knowledge Areas defined by Project Management Institute (PMI), and shall be adopted in the construction project in order to achieve project objectives. As to 4D BIM, it combines 3D BIM model and an additional dimension, namely time data to manage the scheduling issues [82] [83]. Not only does 4D BIM manage the scheduling tasks but also it visualizes the sequential project progress [84]. Consequently, 4D BIM technology improves control over clash detection and changes during the course of the project when the visualized information showing sequential development of the project is analyzed.

Completing the projects within the expected cost is one of the significant criteria for the success of any construction project [85]. Traditionally, the cost controlling process is often conducted manually with a high probability of human error, and the more complex the projects become, the more the error rate increases; however, using computer-related applications allows for more reliable cost control decisions [86]. Therefore, projects utilize many tools such as cost control integrated schedule software, Enterprise Resource Planning (ERP) software and 5D BIM software, which are also commonly used in the construction industry to manage the cost during the course of the projects. Since BIM integrates 3D model with time and cost, it provides improved control on the cost of the projects. Moreover, since the scheduling tasks and costs are assigned to the same elements in BIM, the cash flow of the projects can be generated easily. Furthermore, any change in the project can be directly reflected to the 3D graphical model, time data and cost data at the same time. BIM database may also contain different costs received from many vendors assigned to a building element and the user could easily simulate numerous design scenarios to find out the most effective solution [87].

Although there is a consensus about the definitions of BIM dimensions concerning 3D, 4D and 5D, there is a discrepancy for the 6D and 7D related to Sustainability, Facility Management (FM) or Safety. According to a survey held by Charef et al [78], 86% of the practitioners refer to sustainability for the 6D and 85% of the practitioners refer to Facility Management activities for the 7D. Additionally, while some researchers include FM into 6D BIM [87]–[92] others consider FM as 7D BIM [79], [93], [94]. As concluded by Charef et al. "*This lack of clarity on BIM dimensions beyond the 5^th^ dimension leads to the risk to lose the benefits brought by these extra BIM dimensions*." In this thesis, it is adopted that 6D BIM process is linked with FM. 6D BIM contains project life cycle information that facilitates Operation and Maintenance (O&M) phase [88]. 6D model is updated during the construction phase as an as-built model and is handed over to the owner. The model may include manuals, specifications, warranty information and any other related information which are required during the O&M phase. It also enables to capture recorded data during the course of the operation phase, analyze the energy efficiency of a structure or part of the structure, monitor and optimize the lifecycle cost of the structure.

3.2 Features of 4D-BIM Software

Although numerous BIM tools have been introduced, they may provide different features to satisfy the end-users [59]. Since this thesis concentrates on the development of delay analysis method and its integration into a BIM tool, the most adequate software is to be adopted. Therefore, the most popular BIM tools are compared under this section. At the outset, comparison criteria are detected. Eastman et al proposed required features for 4D BIM tools as follows [79];

- Schedule import capabilities: Concerning BIM software, supporting the planning software to import the schedule data is a very time-saving process in terms of working on different platforms for scheduling issues.
- Merge/Update for 3D/BIM building Model: If a project includes models created in more than one BIM software, the 4D modeling process should support transferring and merging these models into a single tool. Capabilities concerning the schedule update for the portion of the model or all of the models ease the construction process.
- Reorganization: After the data is imported, they should be able to be reorganized. Tools that support easy model reorganizing significantly speed up the modeling process.
- Temporary Components: 4D BIM should support adding temporary components like scaffolding and cranes; therefore, 4D BIM requires having a library that allows users to add these components to 4D model. This kind of temporary element needs to be added to the schedule and having this capability is very essential for the scheduling issues.
- Animation: 4D BIM tools should allow the users to simulate the structure for a specific period of time, and this makes it easier to communicate with stakeholders and detect any inconvenient process.
- Analysis: 4D BIM tools should identify the activities to be constructed concurrently.
- Output: 4D BIM should support extracting multiple snapshots or create movies for specific time periods. This feature eases the process to share the data with the concerned project team.
- Automatic Linking: Automatic linking the elements of the structure to the associated schedule activities based on some rules or options is very essential in terms of saving time and effort.

Braimah and Ndekugri (2007) conducted an in-depth literature review and gathered the factors influencing the selection of delay analysis methods. The most significant factors, which are also directly related to scheduling issues, are detected as depicted below [36].

- Records availability: It refers to the accessibility of project data. This factor is already provided by all the BIM applications.
- Baseline programme availability: The baseline program is a reference for the delay computations and availability of the baseline in BIM is vital to conduct a delay analysis.
- Nature of baseline programme: It stands for the format of the baseline schedule like providing a Bar Chart or CPM. The most reliable and accepted delay analyses by the practitioners and courts rely on the CPM.
- Updated programme availability: The availability of the updated schedule is vital to compute the deviation between the planned and actualized schedule.

Bexel Manager software is adopted in this thesis to incorporate the developed delay analysis model by considering the advantages of the 4D BIM tools.

3.3 Necessities of Integrating a Delay Analysis Method into BIM Environment

In the recent past, many universal breakthroughs concerning BIM development have been made. Meanwhile, the current problems faced during BIM implementations by practitioners have become a driving force encouraging researchers to develop better practices. As a consequence of this, identifying the foregoing obstacles as per their significance is assumed to be very crucial to maintain the fragile structure of the construction industry and thus reduce time and cost consumption via BIM. Time and profit, which are the major objectives of the construction projects, require be controlling and monitoring very closely during the course of the projects. Serving as a remedy in this matter, 4D and 5D in BIM help project stakeholders even with insufficient knowledge concerning the construction domain to understand the potential problems and improve the common understanding of schedule and cost via visual communication. Additionally, change is an inevitable part of the construction projects [95], and reflecting the change in design to the other disciplines such as planning and cost is vital to harmonize the disciplines of projects; however, if synchronization is not conducted in real-time but made manually by using importing and exporting tools, harmonization of project disciplines can be disrupted and this results in cost and time overrun. Therefore, it is recommended that, like ERP tools, BIM tools should keep up with the time and synchronize 3D, 4D and 5D together in real-time. Likewise, the difficulties in forensic schedule analysis and the assessment of the Extension of Time (EOT) soar with the complexity of a project [96]. When 4D and 5D BIM are used efficiently in the construction industry, scheduling software will not be needed any longer. In this case, integration of delay analysis with BIM software is compulsory and thus any contribution in this regard is very fruitful for the practical field. Despite the importance of delay analysis, the study conducted by Cevikbas and Isık [97] revealed that existing 4D-BIM software is not integrated with any delay analysis methods, and concluded that developing BIM-based delay analysis software is highly needed.

4 METHODOLOGY

Each project has an objective to be accomplished by an organization with limited resources within a defined period of time. Due to the dynamic and temporary nature of projects involving sophisticated multidisciplinary works [10], [98], planning the synchronization of all these limited resources is extremely complex and this gives rise to uncertainty [99]. Additionally, rapid developments in technology and increase in competition between the construction companies have been accelerating the number of fast-track projects along with associated risks [100]. Consequently, it is highly probable for the projects to encounter errors and omissions in design, construction methods and contract documents, which result in changes in projects [101], [102]. These changes in projects may lead to adverse outcomes for construction projects [1] and may turn out to be delays followed by conflict(s) and claim(s). Changes resulting in delay in projects are one of the major concerns of the construction projects. Therefore, the negative effects of the delays have to be resolved by using appropriate delay analysis methods. Therefore, firstly, this study aims to detect the major problems of the existing delay analysis methods via scientometric analysis and unstructured interviews. Secondly, by considering the drawbacks of the existing delay analysis methods, this study has an objective to develop a new delay analysis method. Lastly, since 4-D BIM implementation has a great potential to boost performances of the construction projects, this study aims to integrate the developed method into a 4D BIM software. The general methodology of this thesis is illustrated in the figure below.

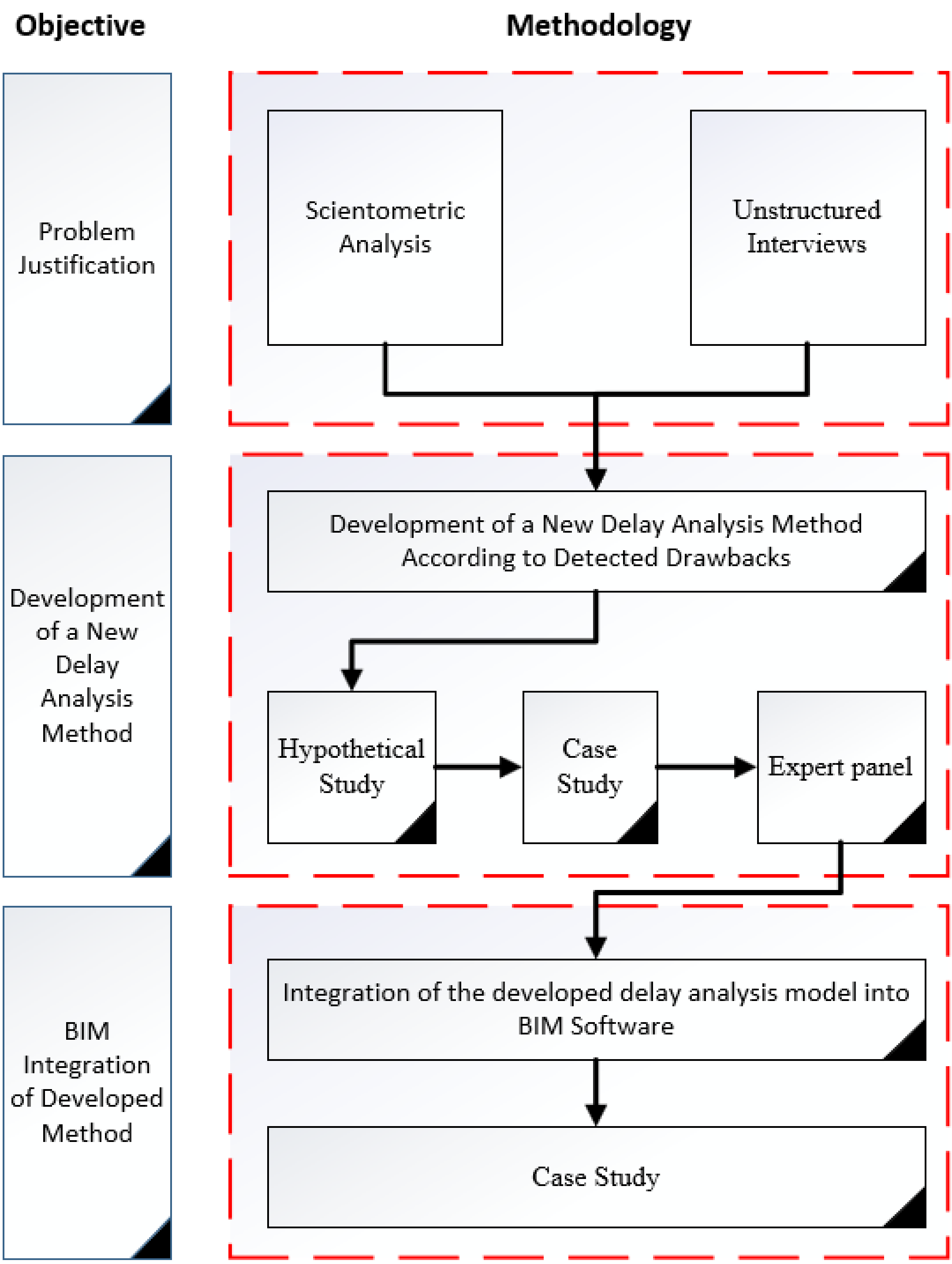


Figure 4.1 Methodology of the thesis

Methodology of this study concerning problem justifications is provided in the sections, namely "Scientometric Analysis" and "Unstructured Interview" respectively. Later, the methodology concerning the developed delay analysis method is presented in the section of "Developing a Numerical Model and Procedure Concerning Delay Analysis". Lastly, the methodology of this study concerning BIM implementation of the developed method is demonstrated under the section, namely “Integration of MSvsMUS into BIM Software as Add-In Tool”.

4.1 Scientometric Analysis

This study adopted science mapping method in order to detect the current trend and gap in the literature as well as the most relevant sources, organizations, authors and countries concerning the delay analysis topic by analyzing and visualizing large volumes of scientific documents. Science mapping, which is a generic process of domain analysis and visualization, targets to detect the intellectual structure of a scientific field [103]. The research methodology consists of three topics, namely ‘Data Collection’, ‘Selection of Science Mapping Tool’, and ‘Scientometric Techniques’. The summary introducing the systematic flow of the scientometric technique adopted in this chapter is schematized in

Figure **4.2** below.

Figure 4.2 Methodology of scientometric analysis

4.1.1 Data Collection

Finding the gaps in the academic area and determining the research tendency is extremely important in terms of shedding light on new studies. Although numerous studies have been conducted concerning delay analyses, an all-encompassing study, which lights the way for future studies, to map the global research of the subject concerning delay analyses has not been conducted so far. In this chapter, the most concentrated research areas and the gap in the literature concerning the delay analysis methods in the construction industry will be determined via scientometric analysis. To achieve this objective, the methodology of this study is defined through the following steps.

4.1.2 Selection of Research Database

Many databases can be preferred for mapping the bibliographic data such as Dimensions Database, Google Scholar, Web of Science and Scopus. Google Scholar can cause double citation counts and results in inadequate data for scientometric analysis [104]. Scopus, which is one of the most comprehensive databases, doesn't lead to double counting problems and includes a wider range of the latest publications than Web of Science [105]. Dimensions Database, covers almost %97 of the SCOPUS database and it is a free scholarly database [106]. Considering all the circumstances, SCOPUS Database is adopted to conduct the scientometric analysis concerning delay analyses in construction projects in order to include all the scientific papers in this chapter.

4.1.3 Detection of Keywords

In order to obtain scientific documents that are directly related to delay analyses in construction projects, the most related keywords are defined. Hence, the articles which were cited more than 50 times were listed [6], [9], [14], [34], [73], [76], [107]–[110] with the help of SCOPUS database. When the related articles were elaborated, it was detected that either 'delay analysis' or 'delay claim' as keywords are common in the areas concerning title, abstracts and keywords of the articles. In addition to this, items such as 'project' and 'construction' are also common in all parts of the articles.

4.1.4 Detection of Relevant Documents

When these keywords were searched via the search engine of Scopus, 240 documents published between 1982 and 2020 (as of 15th November) were detected. All the documents were examined to identify the researches concerning delay analysis. Thus, the objectives of unrelated 72 documents were categorized as tabulated in Table 4.1. Consequently, 168 documents remained after the deduction of these documents from the list.

Table 4.1 Categorization of objectives of omitted documents

Objectives of Delay Analysis	**Number**
Not related study	14
Delay factors	12
Cost	9
The topics of proceedings	9
Scheduling	5
Delay claims	3
Repeated study	3
Delay prevention	3
Preventing the construction delay	3
Documentation	3
Change order	1
Progress monitoring	1
Legal risks	1
Exploring critical conflict issues	1
Excusable delays	1
Shared risks	1
Bidding	1
Productivity	1
Total	72

4.1.5 Selection of Science Mapping Tool

In order to examine any scientific field, an appropriate science mapping tool is to be adopted. Although there are many tools such as VOSviewer, Gephi, CiteSpace, Sci2 and HistCite, the tools, namely VOSviewer, Gephi and CiteSpace are the most prominent [111]. Gephi is a free open-source visualization and exploration software for graphs and networks [112]. CiteSpace, which was developed as a tool for visualizing and analyzing trends and patterns in the literature, is also a free science mapping tool [113]. VOSviewer, which is one of the most recommended mapping and visualization tools that can illustrate data in a great visualized form [114], [115], has special features concerning text-mining [116]. After an in-depth elaboration of the concerned software, it was concluded that VOSviewer satisfies the requirements of this study; therefore, the scientific papers obtained through searching with the concerning keywords in Scopus Database were analyzed via VOSviewer software.

4.1.6 Scientometric Techniques

With the help of this quantitative analysis, the current trends towards delay analysis topic in the construction industry, the gap in the literature, potential research areas, the most relevant countries, sources, organizations and authors were revealed via VOSviewer software in this chapter. To achieve this, as a scientometric technique, 'Mappings Based on Text Data' analysis was conducted. Concerning the text data, the 'Co-Item Analysis' was accomplished to detect the most repeated words under the title, keywords and abstracts in the documents.

4.2 Unstructured Interviews

By performing unstructured interviews, Section 2 is conducted to detect the further drawbacks in addition to drawbacks obtained from Section 1. To achieve this objective, this section constitutes 2 major headings which are research methodology and findings.

Having a great deal of knowledge and skill in a specific field, experts have very crucial roles in identifying problems in a particular subject and leading for innovations. With this notion, in order to spot the major shortcomings concerning delay analysis methods, an expert review meeting with 7 experts – who have a minimum of 4 years of experience in the delay analysis domain - was conducted and common themes of the participants' views are presented in this study. Experts were selected from different firms which were listed in the Engineering News-Record (ENR)'s 2020 Top 250 international contractors. Demographic information of participants is tabulated in the table below.

Table 4.2 Demographic information of experts

Expert #	**Role of Participant**	**Total Experience (Year)**	**Experience in Delay Analysis Methods (Year)**	**Project Budget**	**Project Type**
Expert#1	MEP Planning Chief	7	5	US$1.17bn	Airport Project
Expert#2	Planning Manager	9	6	US$450m	Airport Project
Expert#3	Planning Manager	11	4	606m Euro	Hospital Project
Expert#4	Contracts Manager	18	12	US$300m	Oil & Gas Project
Expert#5	Project Control Manager	14	9	US$4.5bn	Finance Centre Project Project
Expert#6	Technical Office Manager	15	11	US$3bn	Airport Project
Expert#7	Technical Office Manager	10	7	US$400m	US Embassy Project

Drawbacks experienced by the experts were gathered in the expert review meeting. At the outset, problems detected through in-depth literature review were spotted to be experienced by all the experts during the execution of delay analyses in their projects.

4.3 Developing a Numerical Model and Procedure Concerning Delay Analysis

Each project has an objective to be accomplished by an organization with limited resources within a defined period of time. Due to dynamic and temporary nature of projects, planning the synchronization of all these limited resources is extremely complex and this gives rise to uncertainty. Additionally, rapid developments in technology and increase in competition between the construction companies have been accelerating the number of fast-track projects along with associated risks [100]. Consequently, it is highly probable for the projects to encounter errors and omissions in design, construction methods and contract documents, which result in changes in projects [101]. These changes in projects may lead to adverse outcomes for construction projects [1] and may turn to dispute and claim respectively. Although delay analysis methods have been created for systematic assessments of delayed activities to settle the project debates concerning time, it is clearly inferred from this study that detected pitfalls of concerning delay analysis models can lead to some unenviable outcomes in the construction industry. Therefore, in this thesis, it is intended to develop a numerical model, namely 'Modified Schedule versus Modified Updated Schedule (MSvsMUS)' that enables the projects to conduct the delay analysis by overcoming the foregoing weaknesses of the current delay analyses. After the model is developed, it will be tested via one hypothetical and one case study. Later outputs of the case study will be evaluated with the 7 experts to further validate the developed method. Thus, its benefits against the other methods will be highlighted.

This method will be based on the combination of techniques that takes the advantage of the most common and attainable schedules [29], namely Baseline Schedule and Schedule Updates. MSvsMUS will periodically compute the variance between the planned and updated fragnets (Schedule of CDs and EDs) to detect delays and improvements made by the contractor for the CDs and EDs in contrast to the other delay analyses. Additionally, in contrast to other delay analysis methods, MSvsMUS will present a complete procedure and numerical explanation considering the relationships of activities; and this is believed to bring about consistent applications in practice. Moreover, the attribute concerning periodical schedule updates in this method will not only reduce the fluctuations in the critical paths but also consider the actual sequence of work, which will provide more accurate outcomes. Furthermore, since MSvsMUS will include all delay types (EDs, CDs and NDs), it will be capable of computing concurrent delays in accordance with float ownership agreement among the parties. Overcoming all the depicted drawbacks, all these features of the model is claimed to make MSvsMUS the most robust and reliable model in the literature.

In MSvsMUS, since activities on the critical path have the potential to delay the completion date of a project, this method investigates only the critical path of each window. The flow chart of the general principles of MSvsMUS is illustrated below.

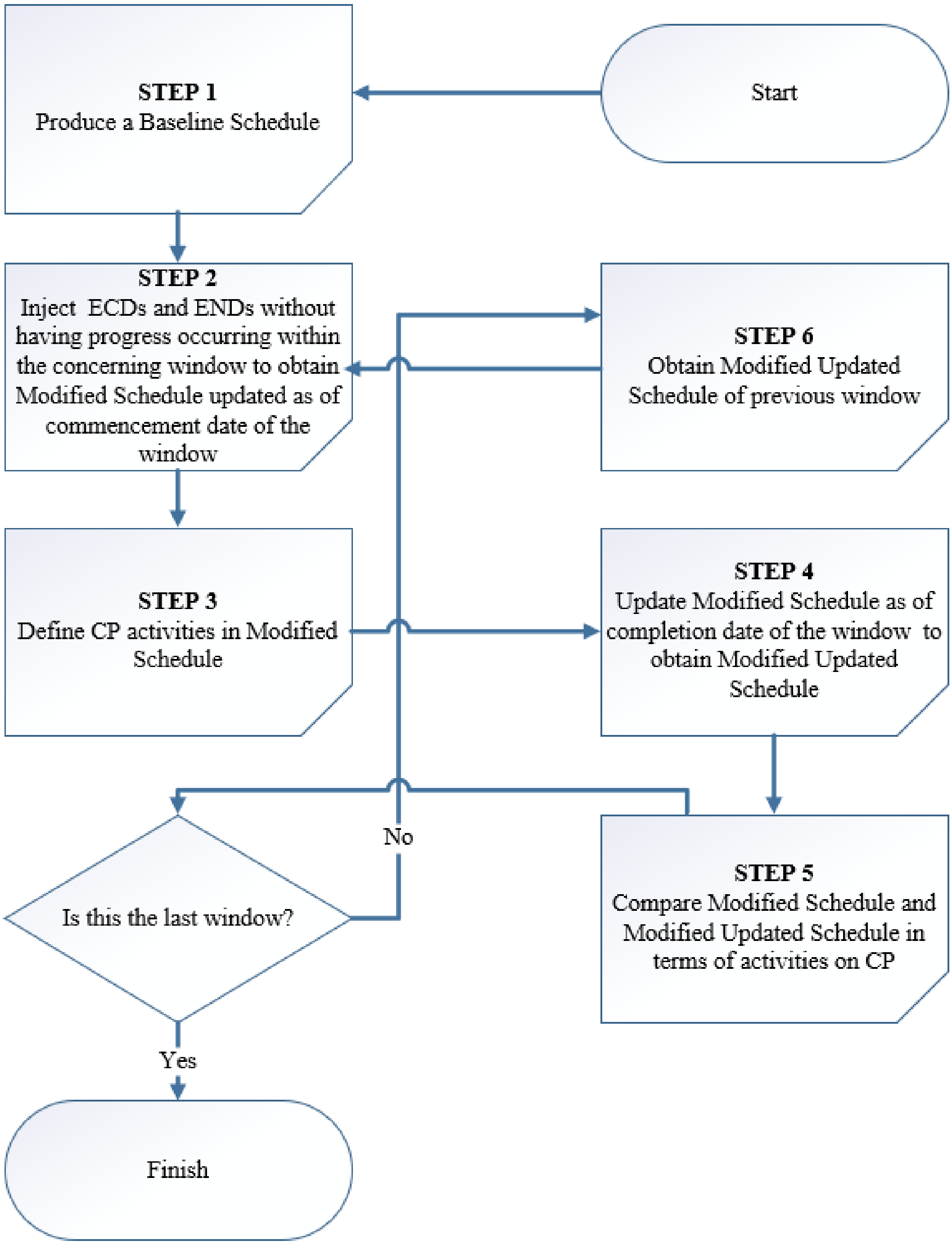


Figure 4.3 Flowchart of the MSvsMUS

Since activities on the critical path have the potential to delay the completion date of a project, MSvsMUS investigates only the critical path of each window. At the outset of the procedure of MSvsMUS, a Baseline Schedule is developed in Step 1. Next, ECDs and ENDs without having progress are injected into baseline or latest schedule update during Step 2 to obtain Modified Schedule (MS) updated as of commencement date of the window. According to Step 3, critical activities are detected in MS. MS is updated as of the completion date of the window to acquire Modified Updated Schedule (MUS) by following Step 4. In contrast to the other delay analysis methods, in Step 5, the technique concerning the comparison of MS and MUS is capable of computing any delays and improvements made by the contractor for the ECDs and ENDs. Moreover, equations defined under step 5 consider the relationships of the activities as opposed to the other delay analysis methods, and this will help to obtain reliable outcomes. Additionally, window-based delay analysis technique of MSvsMUS reduces the fluctuations on CP.

Step 5 - which contains equations from (1) to (21) depicted in the following sections - is conducted under 3 main sections which are 'Early and Late Commencement of Activities', 'Delays or Improvements in Activity Duration', 'Delays under the Owner's Liability (Delays of ECDs and ENDs)'. Any improvements and delays which are under the contractor's liability should be cumulated separately. Delays, improvements and their distributions to the contracting parties derived from each window are summed to obtain the delays of the overall project. In other words, at the end of the project, accumulations of all periods' delays and of the improvements distributed to the responsible parties correspond to the overall delays or improvements of a project. In the equations, summation operator ($\sum$) is an instruction to sum the values obtained from the activities (1 to n) which are on critical path in a defined window period. Additionally, in the following equations, an activity on CP can be either an activity defined in the baseline or a new activity added for ECDs and ENDs; therefore, according to the type of the activity, while left side of the slash mark represents activities existing in the Baseline Schedule, right side of the slash mark represents excusable delays (i.e. ECDs and ENDs) which were added during the project course. Only one side of the slash mark is to be considered at a time, e.g. $US_i/\alpha S_{2i}$.

In the equations presented in this study, “i” is an element of activities within this window.

4.4 Integration of MSvsMUS into BIM Software as Add-In Tool

The number of projects executing scheduling tasks via 4D-BIM has been rising day by day; therefore, 4D-BIM tools including delay analysis are vital to reduce the work effort and the number of employees. In the second phase of this research, the developed delay analysis method will be inserted into BIM software. To be able to integrate MSvsMUS to a 4D BIM software, 4D-BIM software should adopt the Critical Path Method (CPM) to compute the project work schedule. Bexel Manager is one of the most popular 4D-BIM software allowing the user to conduct scheduling tasks through CPM; therefore, Bexel Manager was selected to integrate MSvsMUS.

Application Programming Interface (API) of Bexel Manager permits the developers to access the systems of Bexel Manager on a code level as depicted in Figure 4.4 [117].

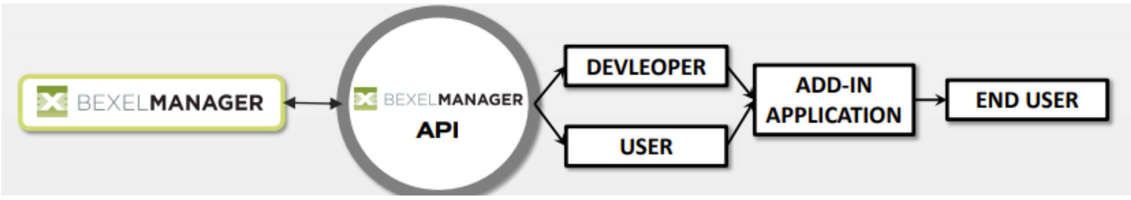


Figure 4.4 Flow chart of development of add-in tool

Bexel Manager allows needs concerning scheduling tasks to be developed as well as many other functionalities via API as illustrated in Figure 4.5 [117].

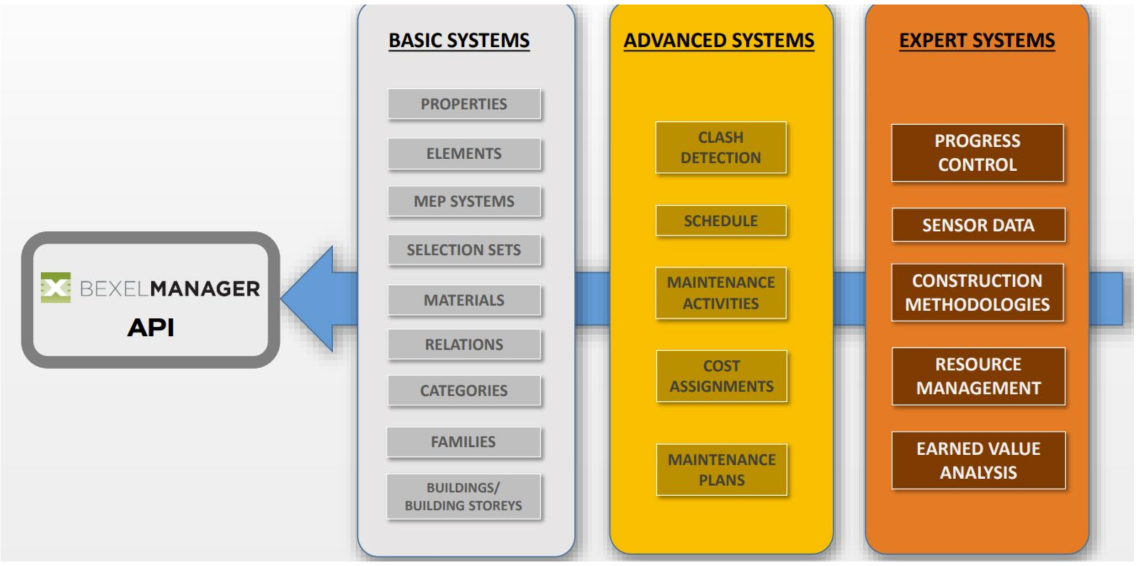


Figure 4.5 Ecosystem of Bexel Manager

Code can be developed through API console or Visual Studio (C#). Pros and cons of API console and Visual Studio (C#) are depicted below [117].

Pros of API console

- Additional tools are not required
- It allows working inside Bexel Manager
- It is suitable for short and fast scripts

Cons of API console

- It is less suitable for more complex solutions

Pros of Visual Studio (C#)

- It is suitable for more complex solutions
- IntelliSense

Cons of Visual Studio (C#)

- It requires Free Visual Studio which is an additional tool
- It requires programming knowledge

Since developing code for MSvsMUS in Bexel Manager is a very complex solution, plugin tool will be developed with the programming language of Visual Studio (C#) required by Bexel Manager's Application Programming Interface (API).

Firstly, Developing Core Functionality side of the plugin tool will be developed. Next, the User Interface side of the software constituting 3 main steps which are Collect User Data, Integrate User Data with Core Functionality and Display Results will be developed. The flowchart of the development of the add-in tool is depicted in Figure 4.6 below.

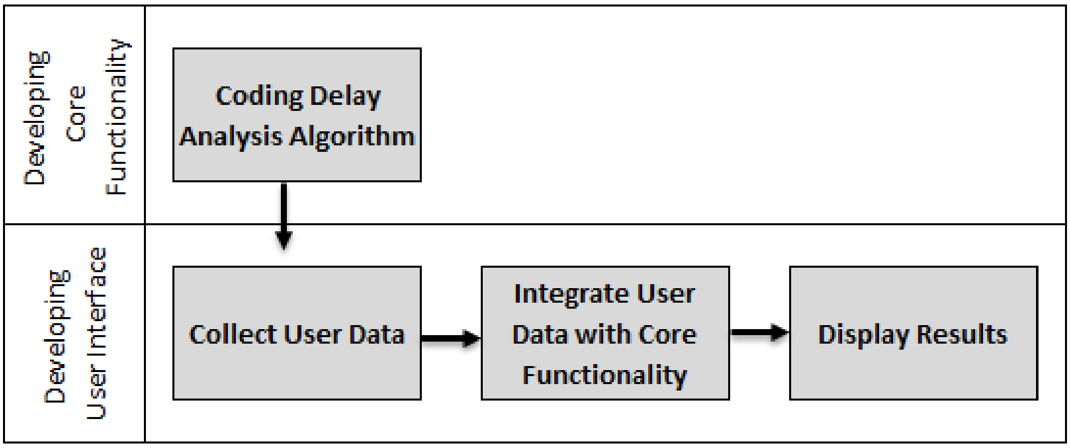


Figure 4.6 Flowchart of the development of the add-in tool

The developed software will be demonstrated in a case study to validate each formulation of MSvsMUS. This method is believed to reduce the project managers' burdens to a great extent through overcoming the current foregoing drawbacks concerning delay analysis.

5

FINDINGS

Resolution of delay related dispute requires a mutual agreement in terms of defining the responsible parties and extension of time. However, identifying responsible parties for delays in construction projects is the most likely source of dispute. Although numerous studies concerning the delay resolution in the construction industry have been published over the past few decades, the construction industry still suffers from the construction delays [118]; therefore, this thesis was conducted in order to improve delay resolution process. Findings of this study concerning problem justifications are provided in the sections, namely "Scientometric Analysis" and "Unstructured Interviews" respectively under this section. Later, the developed method concerning delay analysis is represented under the section of "Developing a Numeric Model and Procedure Concerning Delay Analysis" in this section. Lastly, BIM implementation of the developed method is demonstrated under the section, namely "Integration of MSvsMUS into BIM Software as Add-In Tool".

5.1 Scientometric Analysis

By mapping the term 'Co-Item' based on text data, the most repeated keywords were analyzed via VOSviewer. According to the counting method, full counting was selected to identify occurrences of a term in all documents [119]. Afterwards, the minimum number of occurrences of a term was chosen as 20 in order to detect the most significant keywords and then the meaningful keywords were selected to identify the trend of delay analysis subjects in the construction industry. The most relevant repeated items in documents are illustrated in Figure 5.1 below.

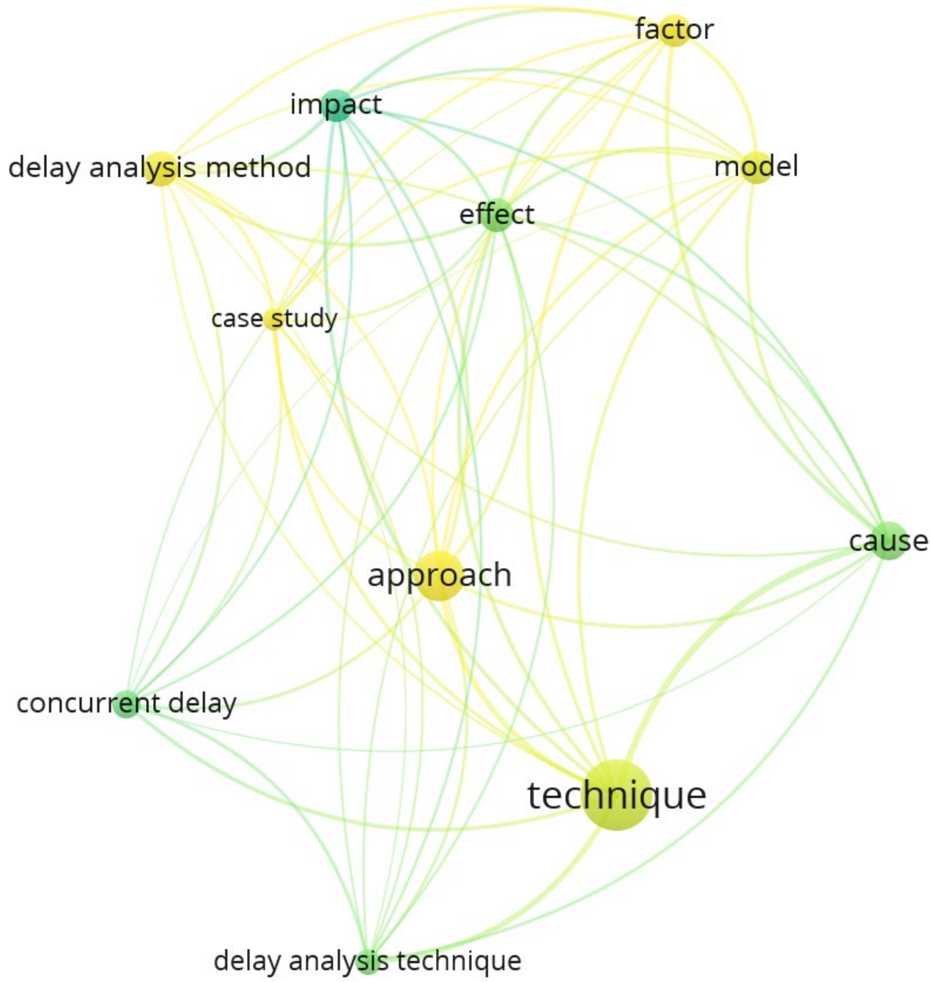


Figure 5.1 Mapping the items repeated in documents

The colours of the items represent the average years of the items as per the repetition in years. While the green ones identify the oldest items, the yellow stands for the newest items. The researches who contributed to the documents coexisting with the items like 'cause', 'effect', 'impact' and 'factor' concentrated on resolving the dispute before it occurred [5], [58], [120]–[133]. Then, the documents spotting the items of 'concurrent delay' came into prominence. Because concurrent delay is a vital issue required to be solved by an appropriate delay analysis method, many researchers have focused on this subject [2], [8], [14], [19], [20], [26], [33], [73], [121], [134]–[142]. When the item of 'case study' is elaborated with the associated documents, it can be concluded that the researchers conducting case studies mostly detected the strengths and weaknesses of the current delay analyses. The most cited articles in this manner [9], [14] emphasize the critical drawbacks of the current delay analyses in the construction industry.The latest released documents containing the items such as ‘approach’, ‘technique’, ‘model’, ‘delay analysis method’, ‘delay analysis technique’ emphasized the pitfalls of the current delay analyses; therefore, they developed new delay analysis models [2], [9], [24], [26]–[30], [33], [34], [137]. In addition to mapping the occurrence of the terms, their magnitudes and subject areas are tabulated in Table 5.1 below.

Table 5.1 Mapping the items repeated in papers

ID	Items	Subject of the Majority of Related Studies	Occurrences	Research Area	Rate
1	Technique	Techniques of delay analysis	126	Improving the delay analysis methods	0.66
2	Approach	Developing delay analysis method	89		
3	Delay analysis		63		
4	Model		59		
5	Concurrent delay	Concurrent delay of owner and contractor	49		
6	Delay analysis technique	Techniques of delay analysis	45		
7	Case study	Strengths and weaknesses of delay	39		
8	Cause	Reasons for delay in construction projects	68	Resolving the dispute before its arising	0.34
9	Factor	Reasons for delay in construction projects	59		
10	Effect	Effects of delaying events	61		
11	Impact		59		

Synthesizing the most cited articles identifies the trend and future directions of the subject concerning delay analysis. This also sheds light on the gaps that need to be improved and new research areas in line with the current needs concerning delay analysis. As depicted in Table 5.1, study areas can be categorized under two sections, which are improving the delay analysis techniques and resolving the dispute before it occurs. These issues are the most relevant topics which constitute the current trend and gap in the literature.

5.2 Unstructured Interviews

Unstructured interviews were conducted with 7 experts and the main themes and the supporting quotations are represented in this section. Mainly four major drawbacks were detected for the existing delay analysis methods which are "Computing any Further Delay and Improvement Made by the Contractor on the Activities of Fragnet", "An Overwhelming Number of Requirements of the Delay Analyses", Lacking the Procedure of the Delay Analysis" and "Considering Relationships of the Activities During the Computation of Delay" as elaborated below.

5.2.1 Computing any Further Delay and Improvement Made by the Contractor on the Activities of Fragnet

With respect to the determination of drawbacks concerning prospective and retrospective delay analyses, an opinion of the Expert#1 that emerge as a result of unstructured interviews is as follows; "*most of the time, the contractor fails to compute the variance between the planned fragnet and actual dates of the fragnet since mostly one of the methods such as prospective or retrospective is adopted in a project. Consequently, for the projects adopting a retrospective delay analysis, it can be claimed by the owner that actualized fragnet was completed in a very abundant time and delayed the project unnecessarily. With respect to projects adopting the prospective delay analysis, issues such as whether the planned fragnet which is on CP will be on the critical path in the schedule update or not, presence of concurrent delays of the contractor occurring with actualized fragnet, determination of mitigation or further delays made by the contractor during the* *schedule update can be questioned by the owner. These issues about TIA and WA can end up with a dispute between the owner and the contractor*". Giving an example from expert#1's previous project, he continued his words as follows; "*The Contractor received 450 Site Instructions during the course of the project. To compute the EOT, Windows Analysis Method was conducted. However, the owner didn't agree upon any of the delay analyses until the end of the project by claiming that duration and sequence of fragnet delayed the project more than required and the contractor didn't take any mitigation measurement on the fragnet.*"

Expert #2 explained the problem arising from the prospective analysis applied in his project with the following words: "*there was an interface between the two main contractors. Time Impact Analysis was conducted to include the interface issues to the schedule and concluded with an impact of 220-days on the project. However, the owner and the Engineer were not convinced by the validity of fragnet and its impact. Settling this debate concerning fragnet took 4 months by conducting several meetings with a remarkable effort. If the type of delay analysis is not defined in the project contract, the owner quite often wants to see the possible effect of the fragnet on the project and actual status of the fragnet. Therefore, the employer can take necessary precautions in advance thanks to the prospective analysis. Besides, the owner may also want to see the retrospective delay analysis since the fragnet which has a time impact in the prospective delay analysis may not be on the CP during the schedule update due to the contractor's concurrent delay.*"

Expert#3 also highlighted a problem experienced in the prospective delay analysis by stating that "*According to PPP contract of the Main Contractor, the project can be delayed a maximum 6 months without reducing the operational period. During the course of the project, the owner sent a change order concerning the increase in the closed area by 14%. The contractor requested 1 year EOT through conducting TIA and requested not to reduce the operation period since the delay in project resulted from change order. However, the owner didn't accept this request of contactor by claiming that the contractor's actual delay or mitigation was not reflected in EOT.*"

Expert #4 made the following remarks on the prospective delay analysis "*The owner tends to question the prospective analysis by claiming that it doesn't reflect the actual concurrency and actual mitigations. When the fragnet becomes actual, the contractor can conduct an additional analysis by comparing TIA with schedule update including the fragnet. However, this additional analysis requires additional efforts of experts and new correspondings between the contractor and the owner.*"

Time Impact Analysis (TIA) and Windows Analysis (WA) are the most acceptable delay analysis methods adopted by practitioners and courts. TIA method may follow up on the project day-by-day from beginning to completion date [14]. However, daily tracking of TIA requires plenty of time, cost and expertize. Consequently, in a practical area, TIA is conducted mostly when a delaying event (Owner or acts of God related delays) occurs so as to detect the possible effect of the delay on the completion date [14], [44], [74]. Therefore, TIA is often preferred for prospective delay analyses. In fact, SCL Protocol 2nd Edition revised its protocol to require the use of TIA for prospective analysis (SCL Protocol 1st edition suggested to use TIA for both prospective and retrospective under the section 3.2.11). "*The conclusions of a prospective delay analysis may not match the as-built programme because the Contractor's actual performance may well have been influenced by the effects of attempted acceleration, re-sequencing or redeployment of resources*" [44]. Since TIA doesn't contain actual dates of a concerning delaying event [44], [47], [143], it doesn't include any mitigation, acceleration and/or further delays concerning fragnet. However, "*The Contractor has a general duty to mitigate the effect on its works of Employer Risk Events.*" [44]; therefore, the potential variance between the planned fragnet included in TIA and updated fragnet can be questioned by the owner in order to detect any further delay or improvement made on activities of the fragnet. Comparing the dates of fragnet in TIA with actualized fragnet opens up an opportunity for overcoming this drawback. However, this method requires additional analysis and consequently much more time, cost and expertize. Like TIA, WA includes only the actualized delaying event since WA doesn't compute any further delays or improvements made by the contractor on the activities in the planned fragnet [31].

5.2.2 An Overwhelming Number of Requirements of the Delay Analyses

All the experts reached a joint decision that requirements of time, cost and expertise of delay analyses depend upon the delay amount, delay type and type of delay analysis. Expert 5 stated the following about the subject; "Although TIA and WA are very useful to some extent, their requirements such as expertise, cost, document control, close site monitoring, record keeping and time obstruct their usage in practice." Expert #6 underlined that if the numerical model and algorithm of the delay analysis are given, the analyst spends less time as well as spending less cost. Also, literature support this view as that applications of TIA and WA in practice are very demanding in terms of a significant amount of requirements such as time, cost and expertise [14], [34], [37], [44], [143].

5.2.3 Lacking the Procedure of the Delay Analysis

All the experts stated that the delay analysis applications may differ from project to project. Although the owner and contractor can agree upon the type of delay analysis at the outset of the project, it is quite encountered that the parties have different views concerning the implementation of the delay analysis method when the related procedure is not available. Expert#7 emphasized the importance of the procedure with the following words; "Project Scheduling Specification which is part of the contract in our project obliges to use TIA for the delay analysis and provides its procedure. When a delaying event resulting from the employer or a delay resulting from force majeure occurs, we conduct TIA and notify the employer. Since we have a procedure, we did not have any technical problems in terms of time claim. However, I believe that lacking this procedure definitely hinders the process."

5.2.4 Considering Relationships of the Activities During the Computation of Delay

Experts have the joint agreement that the most accepted delay analysis methods such as TIA and WA by the practitioners don't have a comprehensive procedure with numerical explanations and practitioners conduct the analysis without taking the relationships into account. Delaying events are detected by taking the total float of the activities - which are computed according to the relationships of the activities - into consideration. Despite this, delay analysis methods are illustrated via Gantt Charts and/or set of equations. The explanations and validations of the existing studies concerning delay analysis methods consider only FS relationships of the activities without considering the other relationships such as SS and FF, and this situation eventuates in inconsistent applications in projects as well as reducing the reliability of delay analysis methods.

5.3 Developing a Numeric Model and Procedure Concerning Delay Analysis

As is mentioned earlier in Methodology Section, the computation methodology consists of 3 main sections, which are 'Early and Late Commencement of Activities', 'Delays or Improvements in Activity Duration Made by Contractor' and 'Delays under Owner's Liability (Delays of CDs and EDs)'.

In the formulations, summation operator ($\sum$) is an instruction to sum the values obtained from the activities (1 to n) which are on critical path in a defined window period. Additionally, in the following formulations, an activity on CP can be either an activity defined in the baseline or a new activity added for CDs and EDs; therefore, according to the type of activity, one side of the slash mark is to be considered, e.g. $US_i/\alpha S_{2i}$.

5.3.1 Detecting the Driven Critical Activities falling into Window

If an activity is in progress in the updated schedule and total float of the activity equals to 0, activity is considered as $critical_n$.

If driven predecessor of $critical_n$ falls into window and there is more than one predecessor, minimum of the following conditions is to be computed in order to detect the driven predecessor of $critical_n$ which is considered as $critical_{n-1}$.

- If one of the predecessors of the $critical_n$ has FS relation with $critical_n$, following formulation is computed;

$$\mathrm{Scritical_n} - \mathrm{Fcritical_{n-1}} - \mathrm{lag}_{\mathrm{critical}_{(n-1)} \rightarrow \mathrm{critical}_{(n)}} \qquad (5.1)$$

- If one of the predecessors of the $critical_n$ has SS relation with $critical_n$, following formulation is computed;

$$\mathrm{Scritical_n} - \mathrm{Scritical_{n-1}} - \mathrm{lag}_{\mathrm{critical}_{(n-1)} \rightarrow \mathrm{critical}_{(n)}} \qquad (5.2)$$

- If one of the predecessors of the $critical_n$ has FF relation with $critical_n$, following formulation is computed;

$$\mathrm{Fcritical_n} - \mathrm{Fcritical_{n-1}} - \mathrm{lag}_{\mathrm{critical}_{(n-1)} \rightarrow \mathrm{critical}_{(n)}} \qquad (5.3)$$

All predecessors of $critical_1$ to $critical_n$ falling into window are computed with the formulations (5.4) to (5.6).

$Critical_{n-1}$ may have more than one relationship with $critical_n$. The relation having the minimum value computed via formulations below is considered driven relationship between $critical_{n-1}$ and $critical_n$.

The representation of critical activities is given in the figure below.

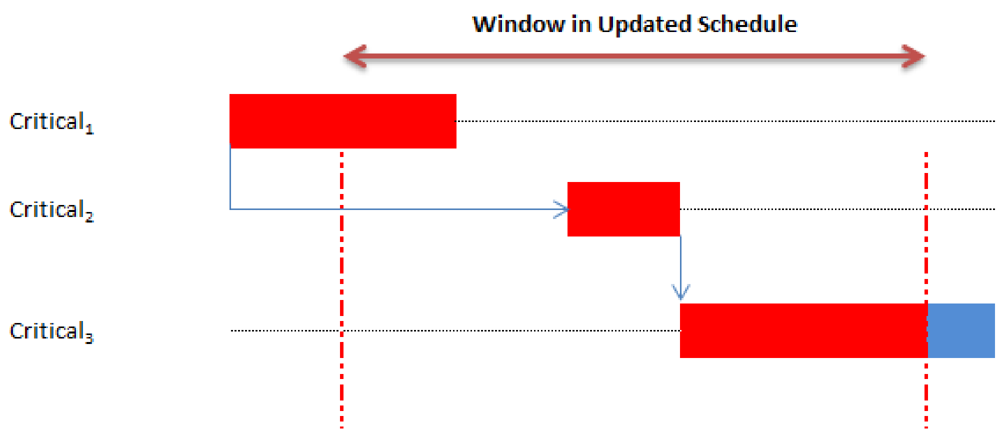


Figure 5.2 Illustration of critical activities in window in updated schedule

If there are more than one driven predecessor, select the predecessor having relationship of FS, FF, SS respectively.

If driven successor of $critical_n$ falls into window and there is more than one successor, minimum of the following conditions is to be computed in order to detect the driven predecessor of $critical_n$ which is considered as $critical_{n-1}$.

- If one of the successors of the $critical_n$ has FS relation with $critical_n$, following formulation is computed;

$$\mathrm{Fcritical_n - Scritical_{n+1} + lag_{critical_{(n+1)} \rightarrow critical_{(n)}}} \quad (5.4)$$

- If one of the successors of the $critical_n$ has SS relation with $critical_n$, following formulation is computed;

$$\mathrm{Scritical_n - Scritical_{n+1} + lag_{critical_{(n-1)} \rightarrow critical_{(n)}}} \quad (5.5)$$

- If one of the successors of the $critical_n$ has FF relation with $critical_n$, following formulation is computed;

$$\mathrm{Fcritical_n - Fcritical_{n+1} + lag_{critical_{(n-1)} \rightarrow critical_{(n)}}} \quad (5.6)$$

All predecessors of $critical_1$ to $critical_n$ falling into window are computed with the formulations provided in following sections.

5.3.2 Early or Late Commencement of Activities

This step aims to compute delays or improvements on the commencement and finish dates of the critical activities that fall within the window period. By taking the relationships of the activities into account, this step distinguishes MSvsMUS from the other delay analysis methods. As a general principle, by taking the relationships between the activities and their predecessor activities into consideration, the value of the gap between the successive activities which are on CP in MUS corresponds to the delay or improvement made by the contractor as represented in the figure below.

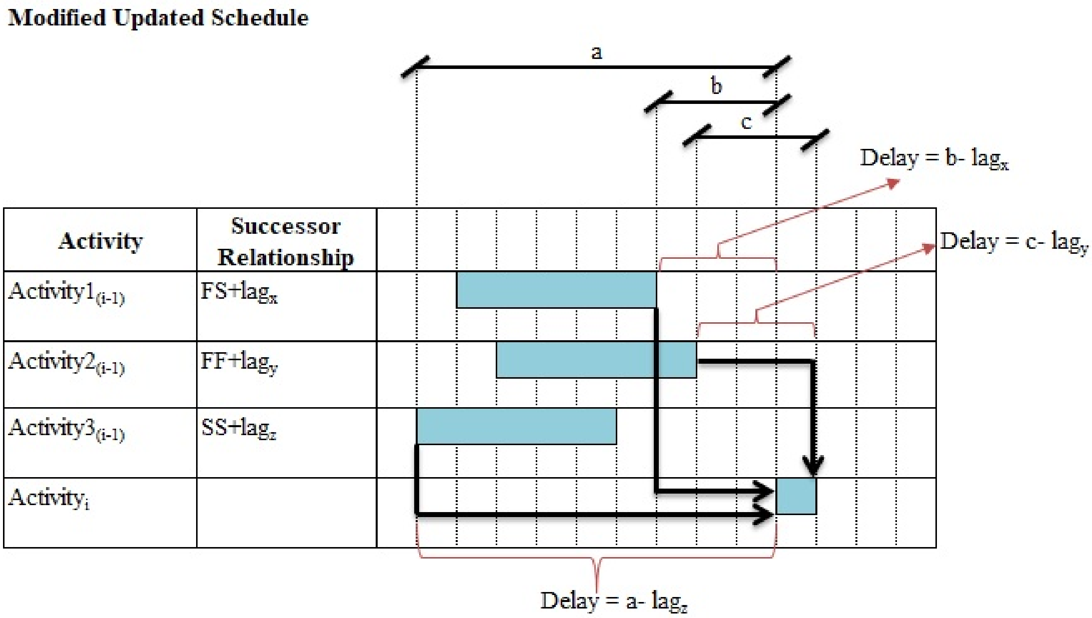


Figure 5.3 Illustration of early and late commencement of activities

The minimum value obtained from the analysis between activity$_{(i)}$ and its predecessor activities is counted as delay and improvement made by the contractor for the concerning window. In each window period, the improvements and delays stemming from the commencement or finish date of activities should be accumulated separately. Unlike other methods, MSvsMUS develops a numeric method by considering the relationships between activities on CP. Only Finish to Start (FS), Finish to Finish (FF), Start to Start (SS) relations are considered in the method as proposed by SCL Protocol (2nd ed.) under Section 1.47.

If an activity$_{(i)}$ which is on CP doesn't have any predecessor and activity$_{(i)}$ starts within the window in MS and it starts within the window in MUS, then delays and improvements of the contractor are obtained by the equation below. While positive values represent improvements, negative values represent delays.

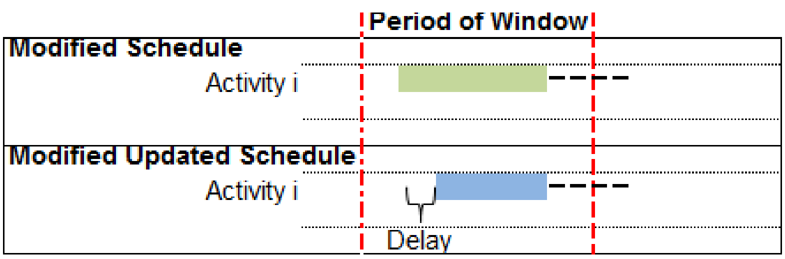


Figure 5.4 Illustration of Formula 5.7 on Gantt Chart

$$\sum_{i=1}^{n}[(MC_i/\alpha S_{1i}) - (US_i/\alpha S_{2i})] \quad (5.7)$$

If an activity$_{(i)}$ which is on CP doesn't have any predecessor and it starts within the window in MS and it starts within one of the following windows in MUS, then the delays of the contractor are obtained for the related window by equation below.

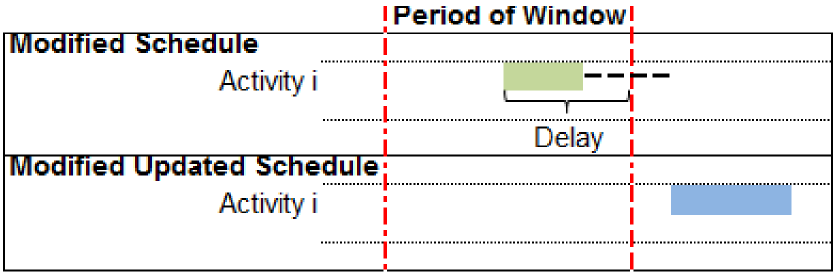


Figure 5.5 Illustration of formula 5.8 on Gantt Chart

$$\sum_{i=1}^{n}[(MC_i\ /\alpha S_{1i}) - \ WFD_i - 1] \quad (5.8)$$

If an activity$_{(i)}$ which is on CP starts within the window in MS and it has driven predecessor/s having FS relationship and the predecessor activity is completed within the window in MS and the activity$_{(i)}$ commences within the window in MUS and driven predecessor activity finishes within the window in MUS, then the contractor's delays or improvements are attained as Equation (5.9). While positive values represent improvements, negative values represent delays.

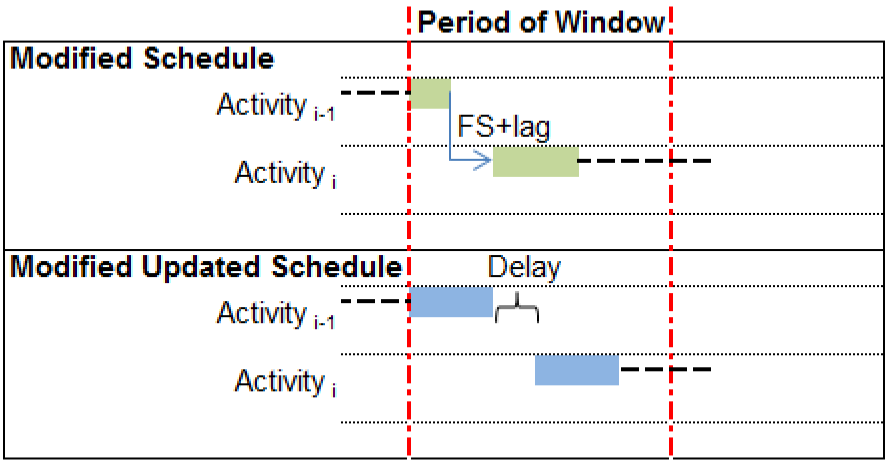


Figure 5.6 Illustration of Formula 5.9 on Gantt Chart (Condition 1)

Additionally, if an activity$_{(i)}$ which is on CP starts within the window in MS and it has driven predecessor/s having FS relationship and the predecessor activity is completed within one of the previous windows in MS and the $lag_{(i-1)\rightarrow i}$ is greater than the equation of $WCD_i - UF_{(i-1)}/\alpha F_{2(i-1)}$ in MUS and the activity$_{(i)}$ commences within the window in MUS, then Equation (5.9) is used to compute the contractor's delay. While positive values represent improvements, negative values represent delays.

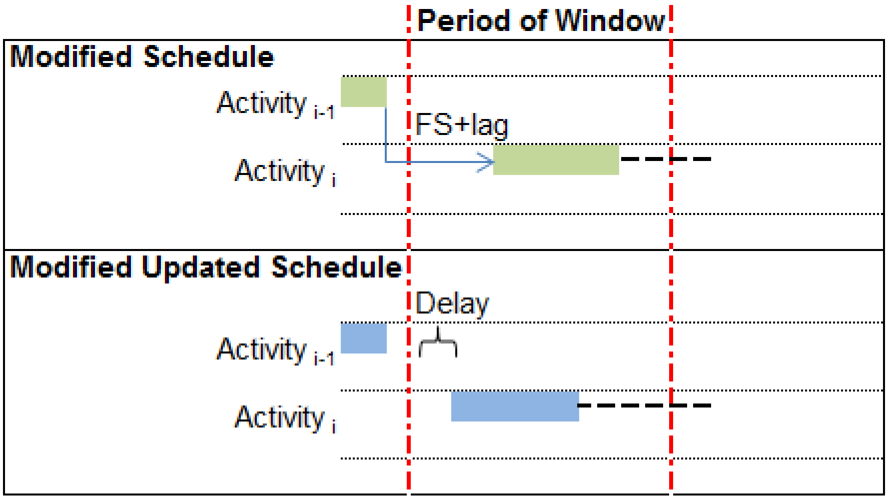


Figure 5.7 Illustration of Formula 5.9 on Gantt Chart (Condition 2)

$$\sum_{i=1}^{n}[(UF_{D(i-1)}/\alpha F_{2D(i-1)}) - (US_i/\alpha S_{2i}) + lag_{D(i-1)\rightarrow i}] \qquad (5.9)$$

If an activity$_{(i)}$ which is on CP starts within the window in MS and it has driven predecessor/s having FS relationship and the predecessor activity is completed within the window in MS and the activity$_{(i)}$ commences within the window in MUS and driven predecessor activity finishes within one of the following windows in MUS, then the contractor's improvements are attained as Equation (5.10).

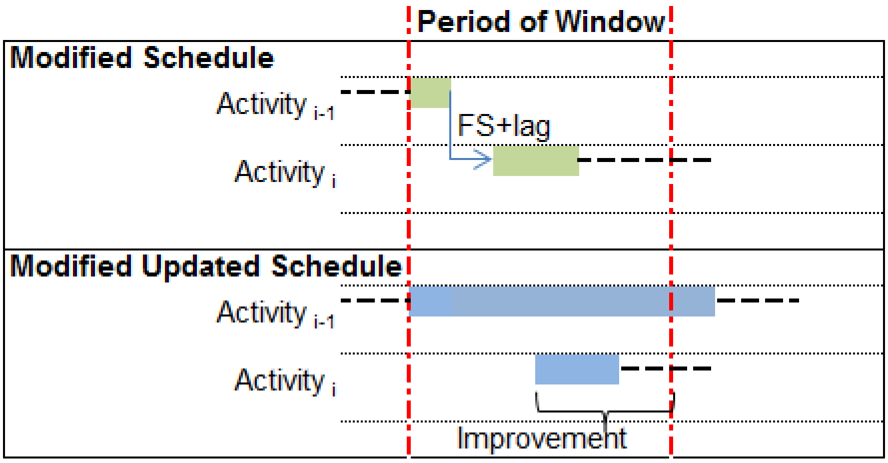


Figure 5.8 Illustration of Formula 5.10 on Gantt Chart (Condition 1)

Additionally, if an activity$_{(i)}$ - which is on CP - starts within the window in MS, and it has driven predecessor/s having SS relationship, and the predecessor activity commences within the window in MS, and the activity$_{(i)}$ commences within the window in MUS, and predecessor activity commenced within one of the following windows in MUS, then the contractor's improvements are attained as Equation (5.10).

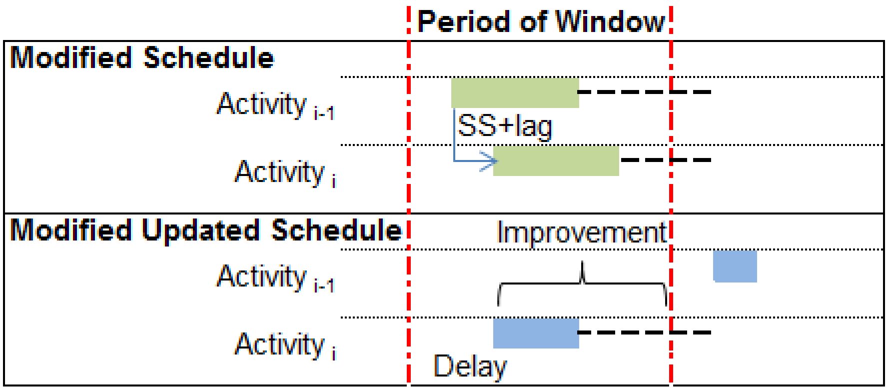


Figure 5.9 Illustration of Formula 5.10 on Gantt Chart (Condition 2)

$$\sum_{i=1}^{n} [(US_i / \alpha S_{2i}) - WFD_i - 1] \qquad (5.10)$$

If an activity$_{(i)}$ has driven predecessor/s with FF relationship and the predecessor activity which is on CP finishes within the window in MS and activity is completed within the window in MS and the predecessor activity is completed within the window in MUS and the activity$_{(i)}$ is completed within the window in MUS, then the contractor's delays and improvements are attained as Equation (5.11). While positive values represent improvements, negative values represent delays.

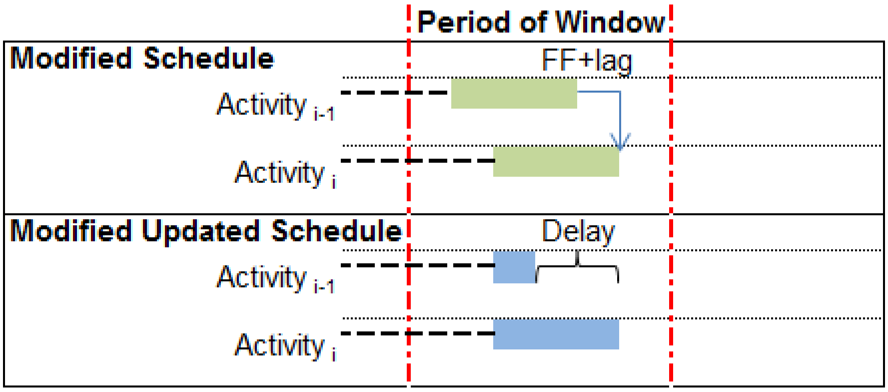


Figure 5.10 Illustration of Formula 5.11 on Gantt Chart (Condition 1)

Additionally, if an activity$_{(i)}$ which is on CP starts within the window in MS and it has driven predecessor/s having FF relationship and the predecessor activity finishes within one of the previous windows in MS and the $\text{lag}_{(i-1)\rightarrow i}$ is greater than the equation of $WCD_i - UF_{D(i-1)}/\alpha F_{2D(i-1)}$ in MUS and the activity$_{(i)}$ finishes within the window in MUS, then Equation (5.11) is used to compute the contractor's delays and improvements. While positive values represent improvements, negative values represent delays.

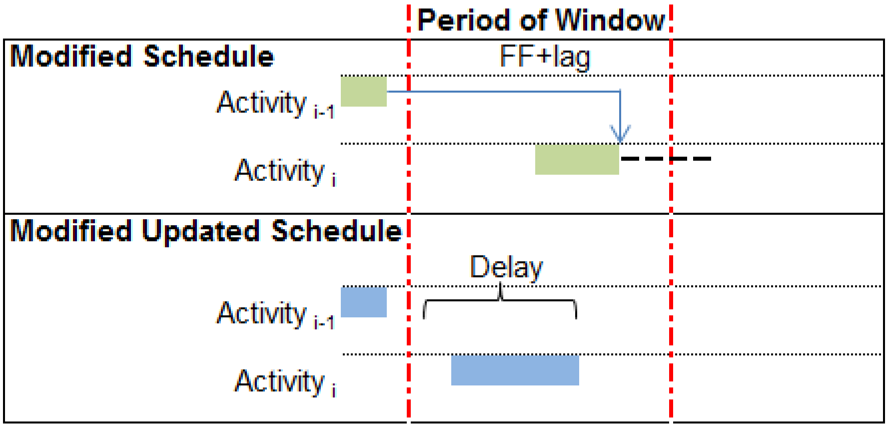


Figure 5.11 Illustration of Formula 5.11 on Gantt Chart (Condition 2)

$$\sum_{i=1}^{n} [(UFD_{(i-1)} / \alpha F_{2D(i-1)}) - (UF_i / \alpha F_{2i}) + lag_{D(i-1) \to i}] \qquad (5.11)$$

If an activity$_{(i)}$ which is on CP finishes within the window in MS and it has driven predecessor/s with FF relationship and the predecessor activity is completed within the window in MS and the predecessor activity is completed within one of the following windows in MUS and the activity$_{(i)}$ is completed within the window in MUS, then the contractor's improvements are attained as Equation (5.12).

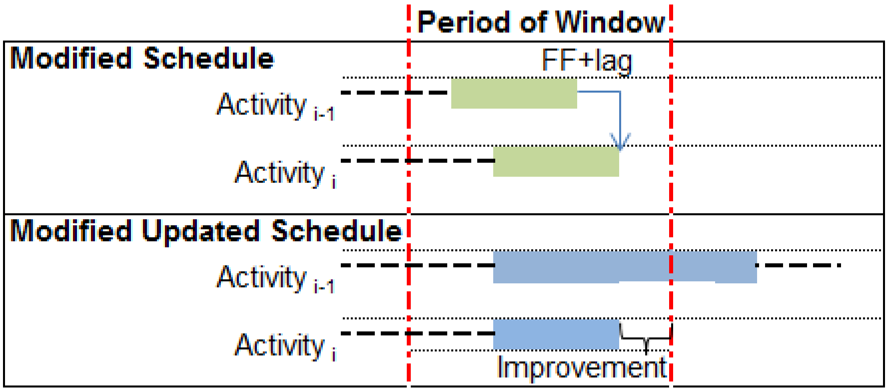


Figure 5.12 Illustration of Formula 5.12 on Gantt Chart

$$\sum_{i=1}^{n} [(UF_i / \alpha F_{2i}) - WFD_i] \qquad (5.12)$$

If an activity$_{(i)}$ which is on CP starts within the window in MS and it has driven predecessor/s having SS relationship and the predecessor activity commences within the window in MS and the activity$_{(i)}$ commences within the window in MUS and predecessor activity commences within the window in MUS, then the contractor's delays and improvements are attained as Equation (5.13). While positive values represent improvements, negative values represent delays.

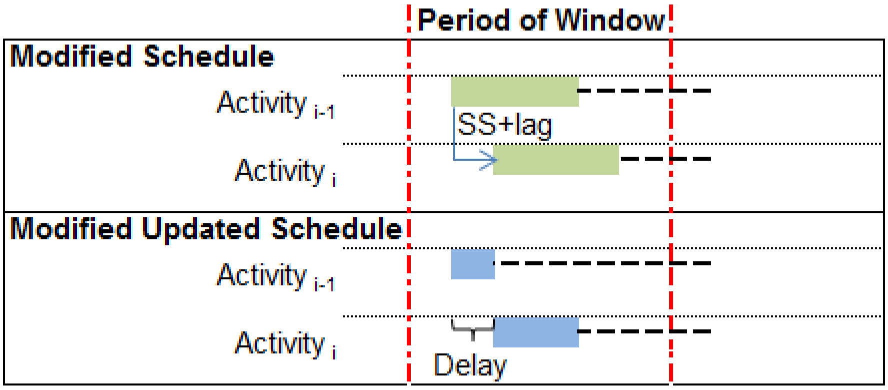


Figure 5.13 Illustration of Formula 5.13 on Gantt Chart (Condition 1)

Additionally, if an activity$_{(i)}$ which is on CP starts within the window in MS and it has driven predecessor/s having SS relationship and the predecessor activity commences in one of the previous windows in MS and the $lag_{D(i-1)\rightarrow i}$ is greater than the equation of $WCD_i - US_{D(i-1)}/\alpha S_{2D(i-1)}$ in MUS and the activity$_{(i)}$ commences within the window in MUS, then Equation (5.13) is used to compute the contractor's delay. While positive values represent improvements, negative values represent delays.

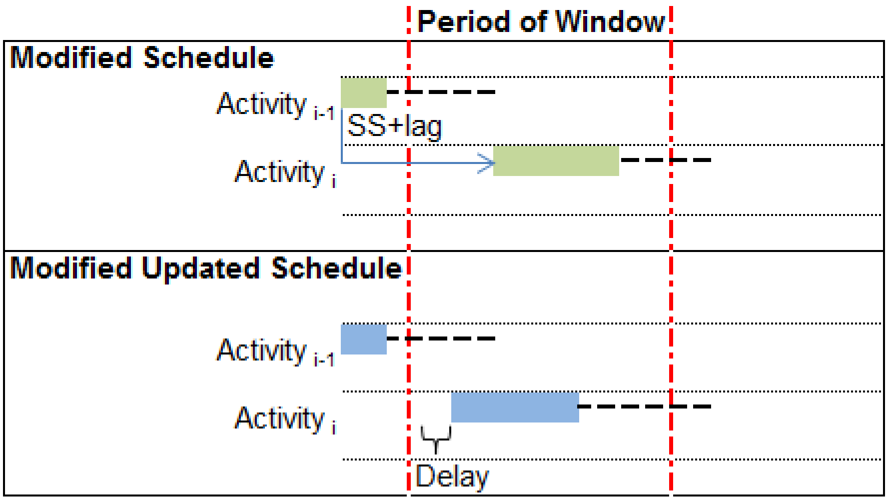


Figure 5.14 Illustration of Formula 5.13 on Gantt Chart (Condition 2)

$$\sum_{i=1}^{n} [(US_{D(i-1)}/\alpha S_{2D(i-1)}) - (US_i/\alpha S_{2i}) + lag_{D(i-1)\rightarrow i}] \quad (5.13)$$

If an activity$_{(i)}$ which is on CP starts within the window in MS and it has driven predecessor/s having FF relationship and the predecessor activity finishes within one of the previous windows in MS and the $\text{lag}_{D(i-1)\rightarrow i}$ is less than the equation of $WCD_i - UF_{D(i-1)}/\alpha F_{2D(i-1)}$ in MUS and the $\text{activity}_{(i)}$ commences and finishes within the window in MUS, then the equation below is used to compute the contractor's delays.

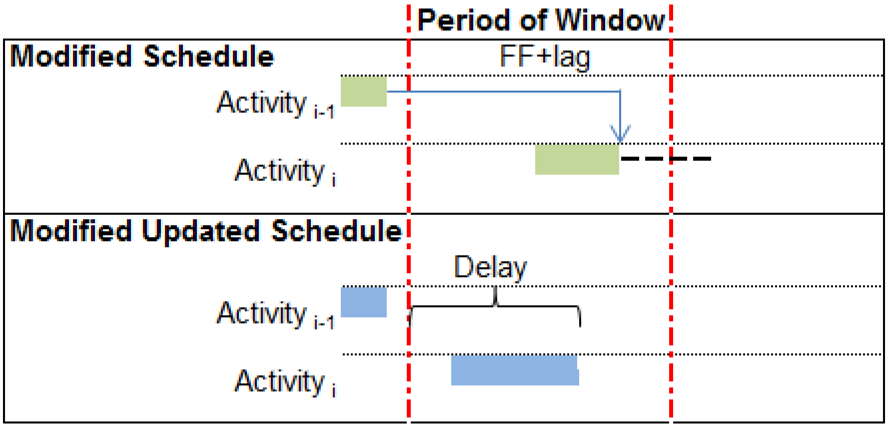


Figure 5.15 Illustration of Formula 5.14 on Gantt Chart

$$\sum_{i=1}^{n}[WCD_i - (UF_i/\alpha F_{2i}) - 1] \qquad (5.14)$$

If an $\text{activity}_{(i)}$ which is on CP starts within the window in MS and it has driven predecessor/s having SS relationships and the predecessor activity is commenced within one of the previous windows in MS and the $\text{lag}_{D(i-1)\rightarrow i}$ is less than the equation of $WCD_i - US_{D(i-1)}/\alpha S_{2D(i-1)}$ in MUS and the $\text{activity}_{(i)}$ commences within the window in MUS, then Equation (5.15) is used to compute the contractor's delays.

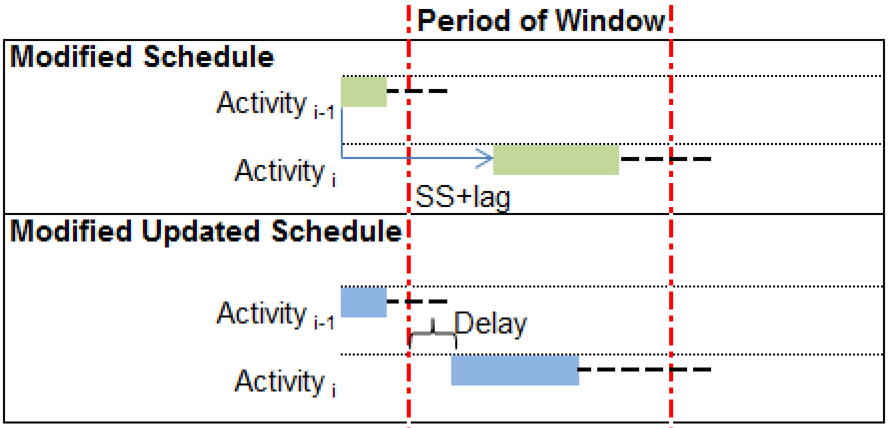


Figure 5.16 Illustration of Formula 5.15 on Gantt Chart (Condition 1)

Additionally, if an activity$_{(i)}$ which is on CP starts within the window in MS and it has driven predecessor/s having FS relationships and the predecessor activity is completed within one of the previous windows in MS and the $\text{lag}_{D(i-1)\to i}$ is less than the equation of $WCD_i - UF_{D(i-1)}/\alpha F_{2D(i-1)}$ in MUS and the activity$_{(i)}$ commences within the window in MUS, then Equation (5.15) is used to compute the contractor's delays.

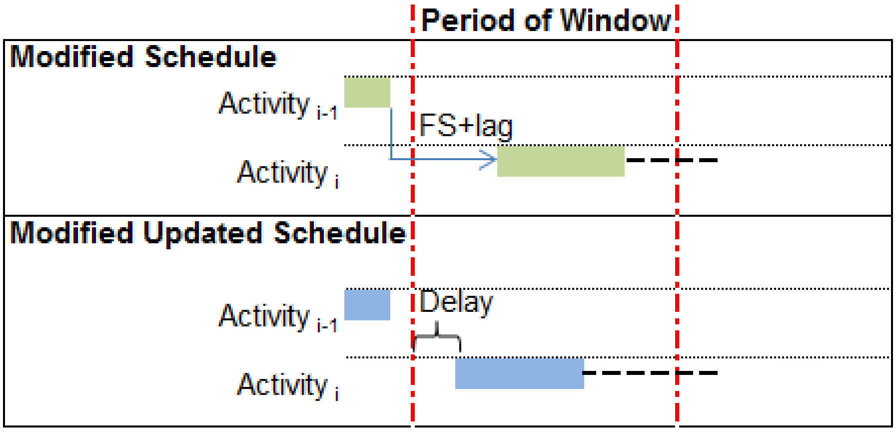


Figure 5.17 Illustration of Formula 5.15 on Gantt Chart (Condition 2)

$$\sum_{i=1}^{n} [\, WCD_i - (US_i/\alpha S_{2i}) \,] \qquad (5.15)$$

If an activity$_{(i)}$ which is on CP starts within the window in MS and it has driven predecessor/s having FS relationship and the driven predecessor activity is completed within the window in MS and the activity$_{(i)}$ commences within one of the following windows in MUS and driven predecessor activity finishes within the window in MUS, then Equation (5.16) is used to compute the contractor's delay. While negative values stand for contractor's delay, positive values are disregarded since there is no definite improvement.

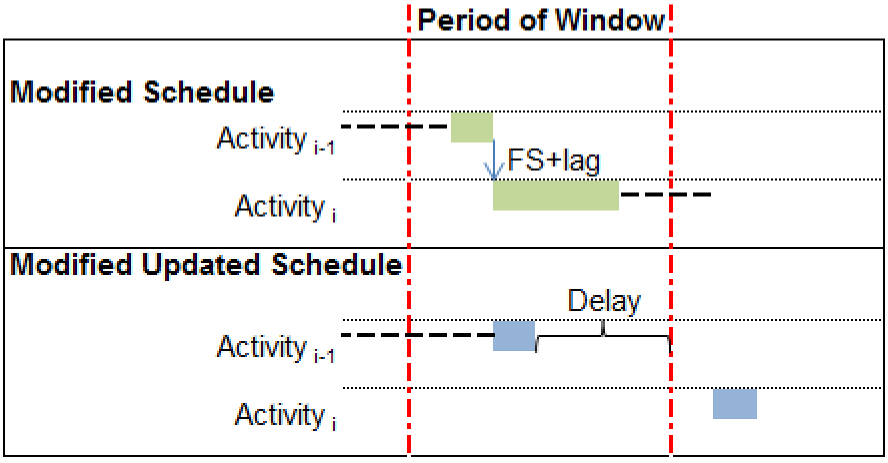


Figure 5.18 Illustration of Formula 5.16 on Gantt Chart (Condition 1)

In addition to this, if an activity$_{(i)}$ which is on CP starts within the window in MS and it has driven predecessor/s having FF relationship and the driven predecessor activity is completed within the window in MS and the activity$_{(i)}$ commences within the windows in MUS and activity$_{(i)}$ finishes within one of the following windows in MUS and driven predecessor activity finishes within the window in MUS, then Equation (5.16) is used to compute the contractor's delay. While positive values stand for contractor's improvement, negative values are disregarded since there is no definite delay.

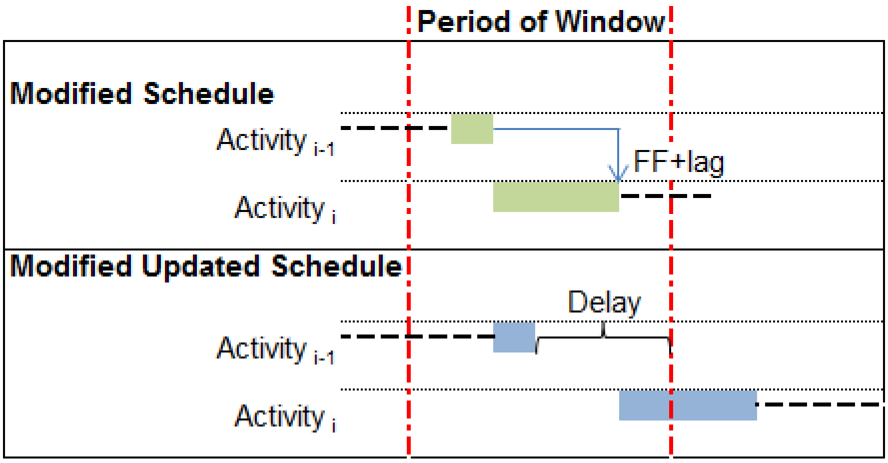


Figure 5.19 Illustration of Formula 5.16 on Gantt Chart (Condition 2)

Moreover, if an activity$_{(i)}$ which is on CP starts within the window in MS and it has driven predecessor/s having FS relationship and the predecessor activity finishes in one of the previous windows in MS and the $\text{lag}_{D(i-1)\to i}$ is greater than the equation of $\text{WCD}_i - \text{UF}_{D(i-1)}/\alpha F_{2D(i-1)}$ in MUS and activity commences within one of the following windows in MUS, then Equation (5.16) is used to compute the contractor's delays.

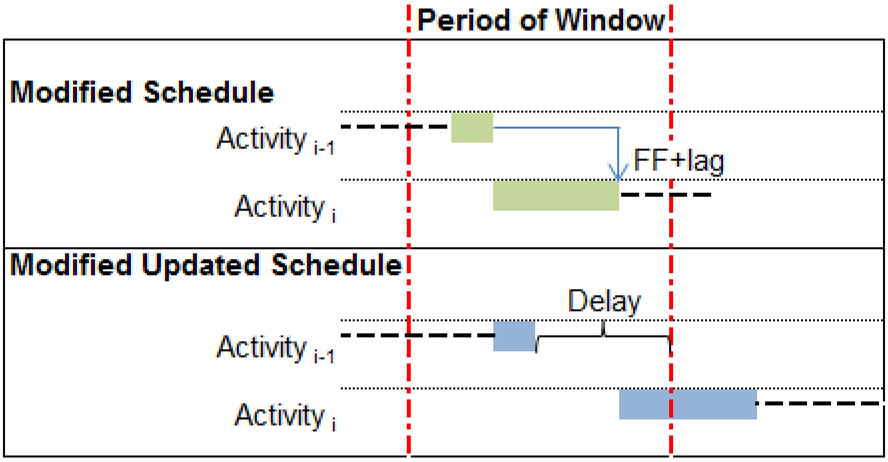


Figure 5.20 Illustration of Formula 5.16 on Gantt Chart (Condition 3)

Furthermore, if an activity$_{(i)}$ which is on CP starts within the window in MS and it has driven predecessor/s having FF relationship and the predecessor activity finishes within one of the previous windows in MS and the $lag_{D(i-1)\rightarrow i}$ is greater than the equation of $WCD_i - UF_{D(i-1)}/\alpha F_{2D(i-1)}$ in MUS and activity finishes within one of the following windows in MUS, then Equation (5.16) is used to compute the contractor's delays.

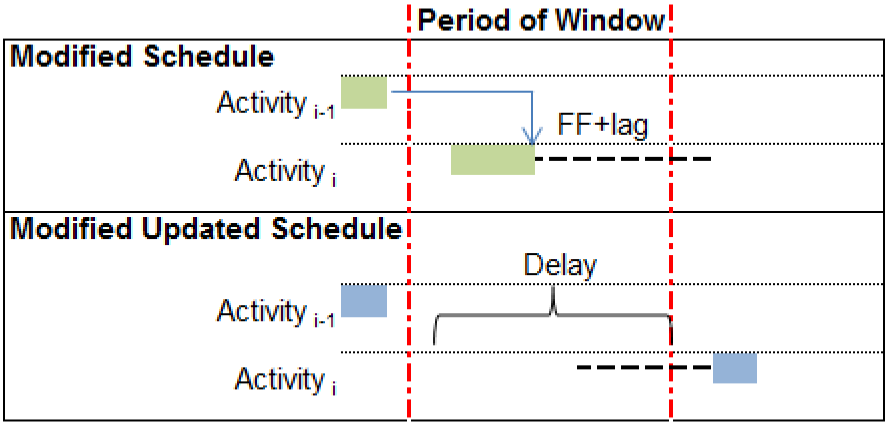


Figure 5.21 Illustration of Formula 5.16 on Gantt Chart (Condition 4)

$$\sum_{i=1}^{n}[(UF_{D(i-1)}/\alpha F_{2D(i-1)}) - WFD_i + lag_{D(i-1)\rightarrow i}] \qquad (5.16)$$

If an activity$_{(i)}$ which is on CP starts within the window in MS and it has driven predecessor/s having SS relationship and the driven predecessor activity is commenced within the window in MS and the activity$_{(i)}$ commences within one of the following windows in MUS and its driven predecessor starts within the window in MUS, then Equation (5.17) is used to compute the contractor's delay. While negative values stand for contractor's delay, positive values are disregarded since there is no definite improvement.

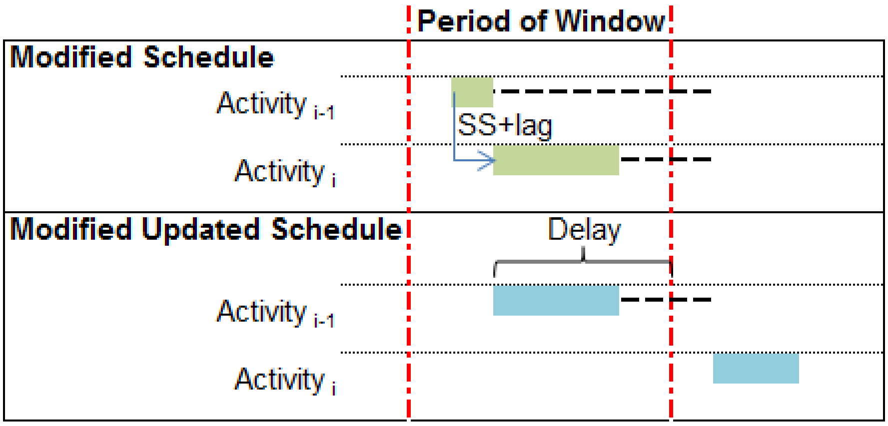


Figure 5.22 Illustration of Formula 5.17 on Gantt Chart (Condition 1)

Additionally, if an activity$_{(i)}$ which is on CP starts within the window in MS, and it has driven predecessor/s having SS relationship, and the predecessor activity commences within one of the previous windows in MS, and the $lag_{D(i-1)\rightarrow i}$ is greater than the equation of $WCD_i - US_{D(i-1)}/\alpha S_{2D(i-1)}$ in MUS, and activity$_{(i)}$ commences within one of the following windows in MUS, then Equation (5.17) is used to compute the contractor's delay.

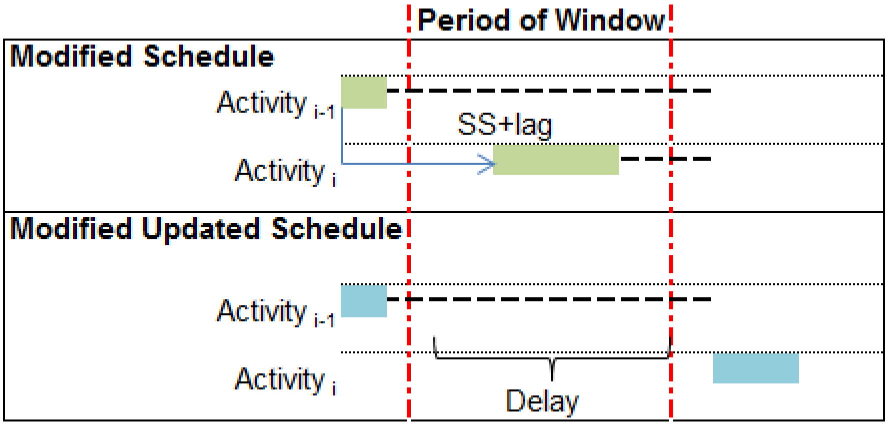


Figure 5.23 Illustration of Formula 5.17 on Gantt Chart (Condition 2)

$$\sum_{i=1}^{n}[(US_{D(i-1)}/\alpha S_{2D(i-1)}) - WFD_i + lag_{D(i-1)\rightarrow i} - 1] \qquad (5.17)$$

If an activity$_{(i)}$ which is on CP starts within the window in MS, and it has driven predecessor/s having FS relationship, and the driven predecessor activity is completed within one of the previous windows in MS, and the $lag_{D(i-1)\rightarrow i}$ is less than the equation of $WCD_i - UF_{D(i-1)}/\alpha F_{2D(i-1)}$ in MUS, and the activity$_{(i)}$ commences within one of the following windows in MUS, then Equation (5.18) is used to compute the contractor's delay.

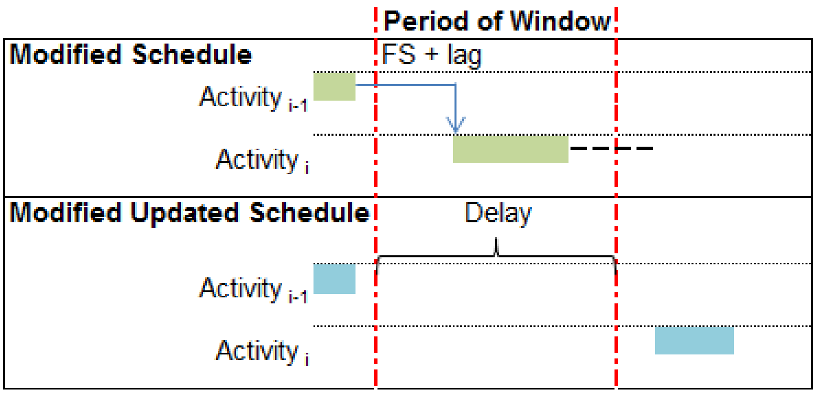


Figure 5.24 Illustration of Formula 5.18 on Gantt Chart (Condition 1)

Additionally, if an activity$_{(i)}$ which is on CP starts within the window in MS, and it has driven predecessor/s having SS relationship, and the driven predecessor activity commences within one of the previous windows in MS, and the $\mathrm{lag}_{D(i-1)\rightarrow i}$ is less than the equation of $WCD_i - US_{D(i-1)}/\alpha S_{2D(i-1)}$ in MUS, and the activity$_{(i)}$ commences within one of the following windows in MUS, then Equation (5.18) is used to compute the contractor's delay.

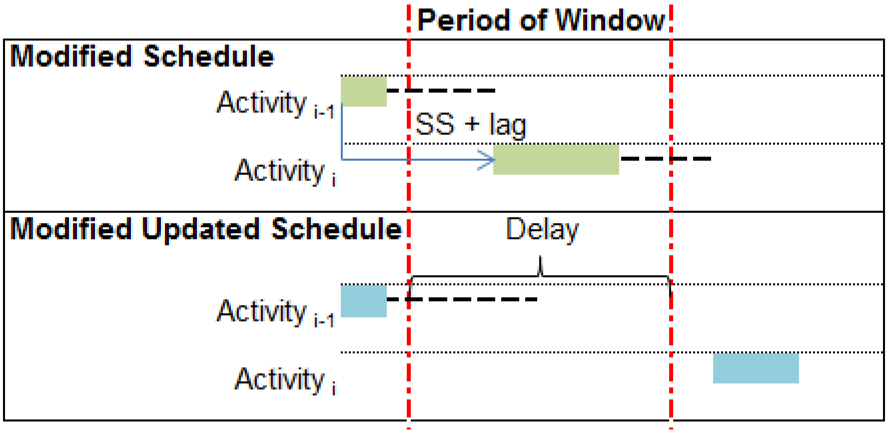


Figure 5.25 Illustration of Formula 5.18 on Gantt Chart (Condition 2)

Moreover, if an activity$_{(i)}$ which is on CP starts within the window in MS, and it has driven predecessor/s having FF relationship, and the driven predecessor activity is completed within one of the previous windows in MS and the $\mathrm{lag}_{D(i-1)\rightarrow i}$ is less than the equation of $WCD_i - UF_{D(i-1)}/\alpha F_{2D(i-1)}$ in MUS, and the activity$_{(i)}$ finishes within one of the following windows in MUS, then Equation (5.18) is used to compute the contractor's delay.

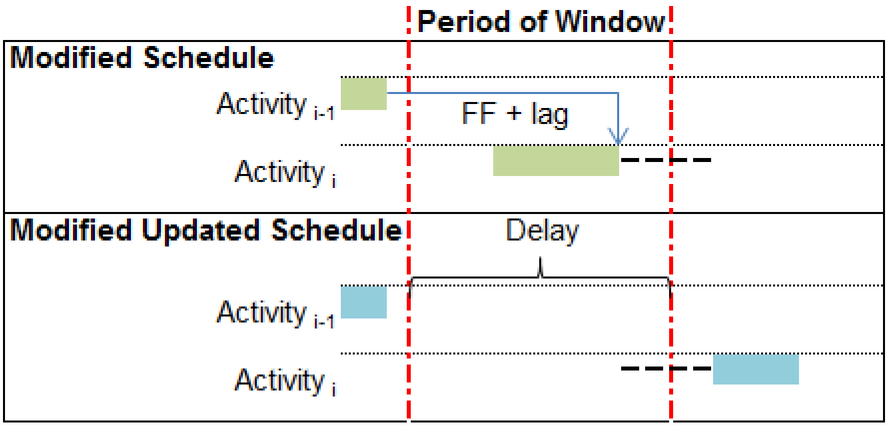


Figure 5.26 Illustration of Formula 5.18 on Gantt Chart (Condition 3)

$$\sum_{i=1}^{n}(\mathrm{WCD_i} - \mathrm{WFD_i} - 1) \tag{5.18}$$

If an $\text{activity}_{(i)}$ which is on CP starts within the window in MS, and it has driven predecessor/s having FS or FF or SS relationship in MS, and the predecessor activity is completed within the window in MS, and the $\text{activity}_{(i)}$ commences within one of the following windows in MUS, and driven predecessor activity commences within one of the following windows in MUS, then no delay is considered for the activity because this activity's delay is already considered during the calculation of the delay of the former driven activity. Additionally, if an $\text{activity}_{(i)}$ which is on CP commences within one of the previous windows in MS and it is in progress or completed within the windows in MS, then no delay is considered for the activity.

5.3.3 Delays or Improvements in Activity Duration Made by Contractor

The goal of this step is to find any actual improvements or further delays made by the contractor on the original durations of activities. Figuring out the delays and improvements made by the contractor for both ECDs and ENDs, which is a part of this step, distinguishes this method from the other delay analyses. In this section, durations of critical activities in MUS are to be compared with the baseline schedule and planned subnet schedule in order to detect any actual improvements and further delays on the original durations of the critical activities falling into each window. Delays and improvements in activity duration are only computed provided that the following conditions are met.

- When an activity which is on CP falls in a window in MS
- When an activity which is in progress or completed in MUS
- When an activity doesn't have a driven successor with SS relation.

Because any delays and improvements in duration logically do not affect the successor activities as illustrated in the figure below.

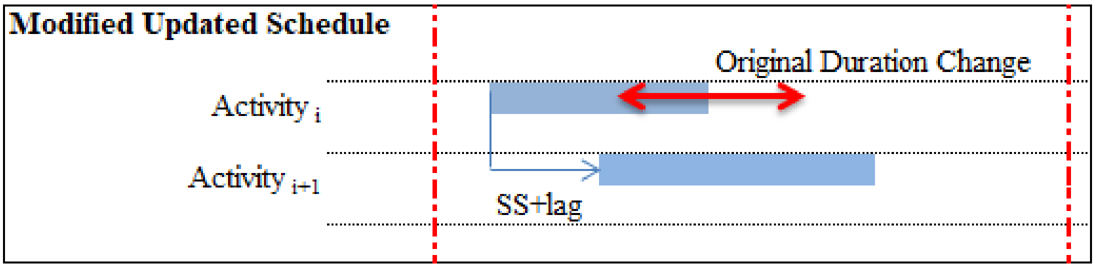


Figure 5.27 Illustration of duration change of an activity having SS relationships with its driven successor

Providing the conditions are met as depicted above, the delays and improvements in durations of activities which are on CP are to be calculated. Improvements and delays concerning the durations of activities are under the contractor's liability and should be accumulated separately. Following the consideration of the statements above, the equations below have been developed to compute delays and improvements made in activity durations by the contractor.

If the activity$_{(i)}$ starts and finishes within the window in MUS, then the equation below is applied. While the positive values represent improvements, negative values represent delays in activity duration.

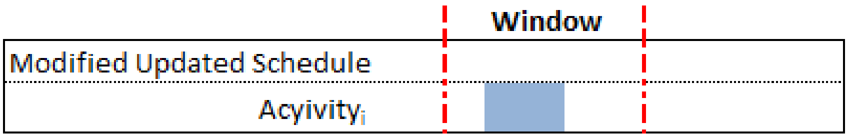


Figure 5.28 Illustration of formula on Gantt Chart

$$\sum_{i=1}^{n} [OD_i/SD_i - ((UF_i/\alpha F_{2i}) - (US_i/\alpha S_{2i}) + 1)] \quad (5.19)$$

If the activity$_{(i)}$ starts within one of the previous windows in MUS and the activity$_{(i)}$ finishes within the window in MUS and the activity$_{(i)}$ finishes within the window in MS, then equation below is applied. While the positive values represent improvements, negative values represent delays in activity duration.

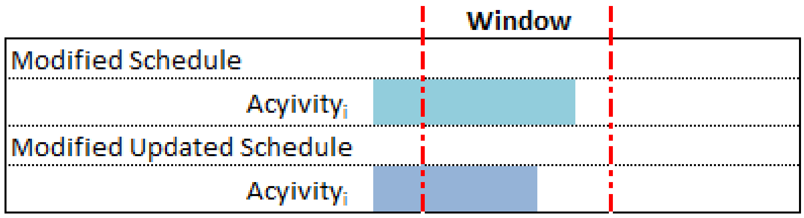


Figure 5.29 Illustration of Formula 5.20 on Gantt Chart

$$\sum_{i=1}^{n} [MF_i/\alpha F_{1i} - UF_i/\alpha F_{2i}] \qquad (5.20)$$

If the activity$_{(i)}$ starts within one of the previous windows in MUS and the activity$_{(i)}$ finishes within the window in MUS and the activity$_{(i)}$ finishes within one of the following windows in MS, then Equation (5.12) is applied. While the positive values represent improvements, negative values represent delays in activity duration.

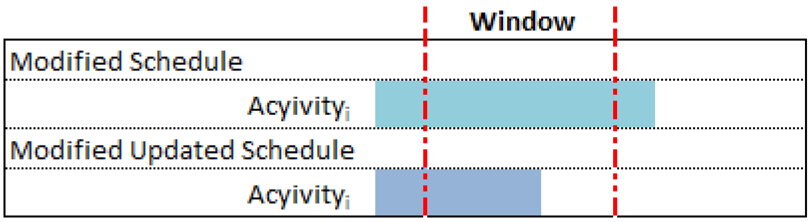


Figure 5.30 Illustration of Formula 5.12 on Gantt Chart

If the activity$_{(i)}$ starts within the window in MUS and the activity$_{(i)}$ finishes within one of the following windows in MUS, then the equation below is applied. While the negative values obtained from the equation are added to the contractor's delays, positive values are not counted because there is no actualized and apparent improvement in activity duration.

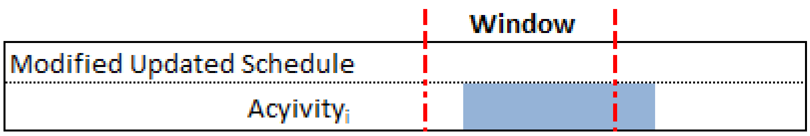


Figure 5.31 Illustration of Formula 5.21 on Gantt Chart

$$\sum_{i=1}^{n} [OD_i/SD_i \; - \; (WFD_i - (US_i/\alpha S_{2i}) + 1)] \qquad (5.21)$$

Additionally, if the activity$_{(i)}$ starts in one of the previous windows in MUS and the activity$_{(i)}$ finishes within one of the following windows in MUS and activity$_{(i)}$ finishes within the windows in MS, then the equation below is applied to detect the contractor's delay.

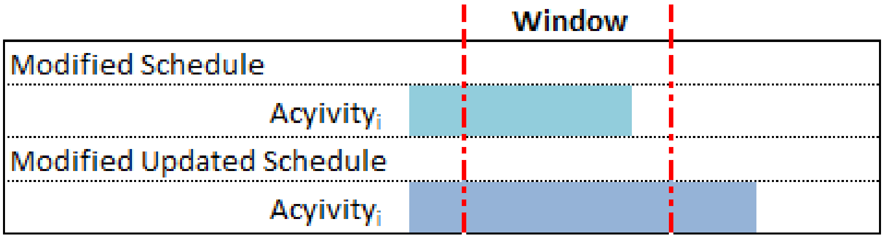


Figure 5.32 Illustration of Formula 5.22 on Gantt Chart

$$\sum_{i=1}^{n} [(MF_i/\alpha F_{1i}) \; - \; WFD_i] \qquad (5.22)$$

If the activity$_{(i)}$ starts in one of the previous windows in MUS and the activity$_{(i)}$ finishes within one of the following windows in MUS and the activity$_{(i)}$ finishes within one of the following windows in MS, then no delays or improvements are considered.

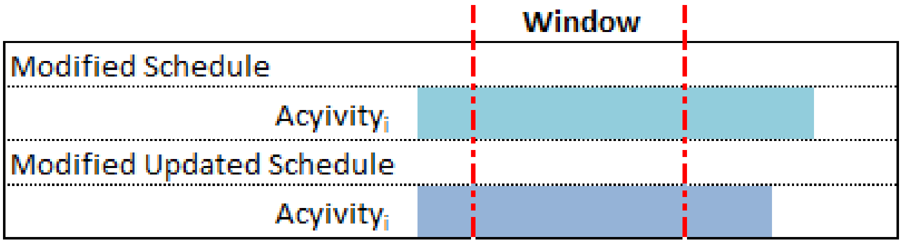


Figure 5.33 Illustration of no delays or improvements in activity duration on Gantt Chart

5.3.4 Delays Under Owner's Liability (Delays of CDs and EDs)

In this section, activities of ECDs and ENDs falling into each window in MS are computed in order to detect the delays which are under the owner's responsibility.

Delays resulting from ECDs and ENDs are only computed provided that the following conditions are both met.

- When ECDs and ENDs are on CP in the concerning window in the MS
- When ECDs and ENDs are in progress or completed in MUS

On condition that the statements above are provided, delays resulting from durations of ECDs and ENDs falling into each window in MS are computed as the owner's delays. Following the consideration of the statements above, equations below have been developed in this section to compute the delays under the owner's liability.

In MS, if ECDs and ENDs don't have any predecessor and ECDs and ENDs commence within the window and driven successor activity commences within one of the following windows, then the delays resulting from ECDs and ENDs are computed via Equation (5.23).

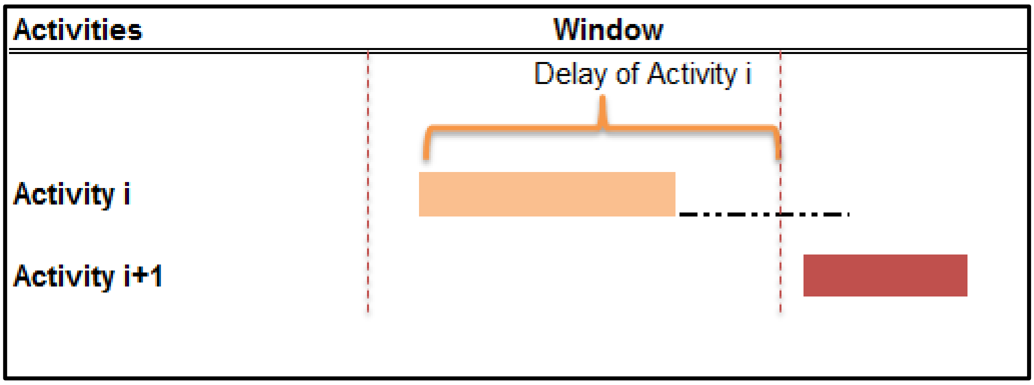


Figure 5.34 Illustration of Formula 5.23 on Gantt Chart

$$\sum_{i=1}^{n} (MC_i - WFD_i - 1) \tag{5.23}$$

In MS, if ECDs and ENDs don't have any predecessor and ECDs and ENDs commence within the window and driven successor activity commences within the window, then the delays resulting from ECDs and ENDs are computed via Equation (5.24).

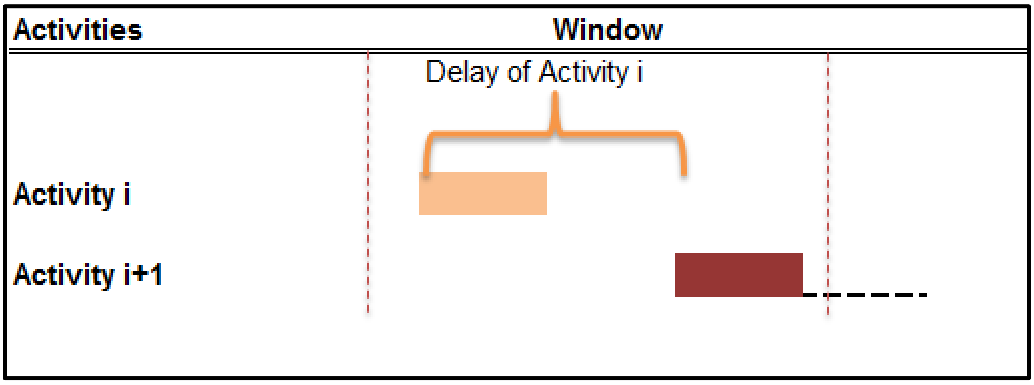


Figure 5.35 Illustration of Formula 5.24 on Gantt Chart

$$\sum_{i=1}^{n} (MC_i - MC_{D(i+1)}) \qquad (5.24)$$

In MS, if ECDs and ENDs start within the window and driven predecessor activity finishes within the window and driven successor activity starts within one of the following windows, then the delays resulting from ECDs and ENDs are computed with Equation (5.25).

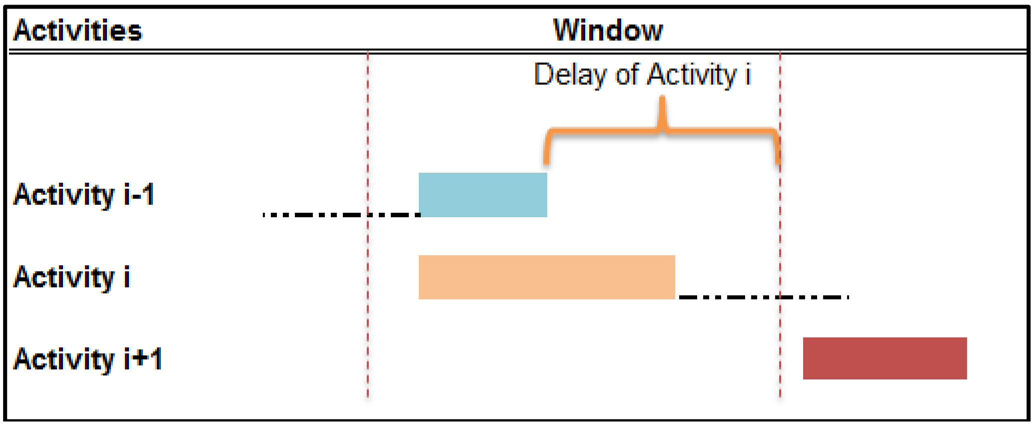


Figure 5.36 Illustration of Formula 5.25 on Gantt Chart (Condition 1)

Additionally, in MS, if ECDs and ENDs start within one of the previous windows and driven predecessor activity finishes within the window and driven successor activity starts within one of the following windows, then the delays resulting from ECDs and ENDs are computed with Equation (5.25).

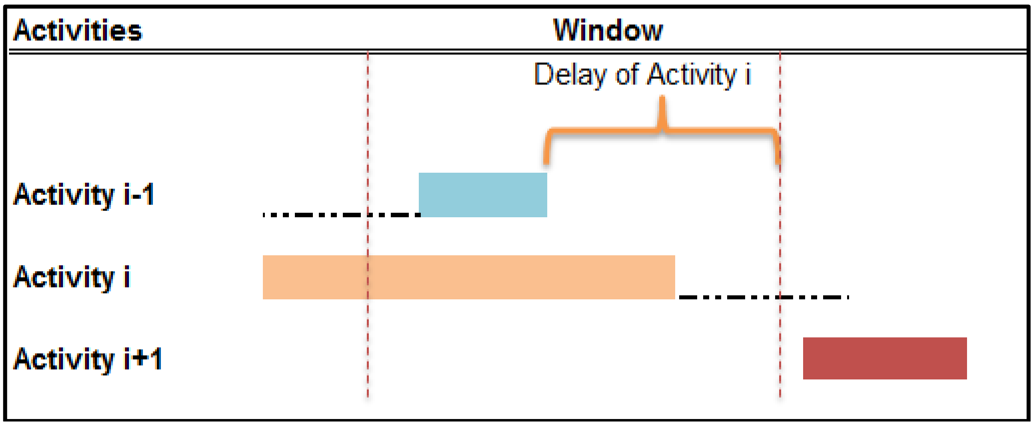


Figure 5.37 Illustration of Formula 5.25 on Gantt Chart (Condition 2)

$$\sum_{i=1}^{n}(MF_{D(i-1)} - WFD_i) \tag{5.25}$$

In MS, if ECDs and ENDs start or complete within the window and driven predecessor activity finishes within one of the previous windows and driven successor activity starts within one of the following windows, then the delays resulting from ECDs and ENDs are computed with Equation (5.26).

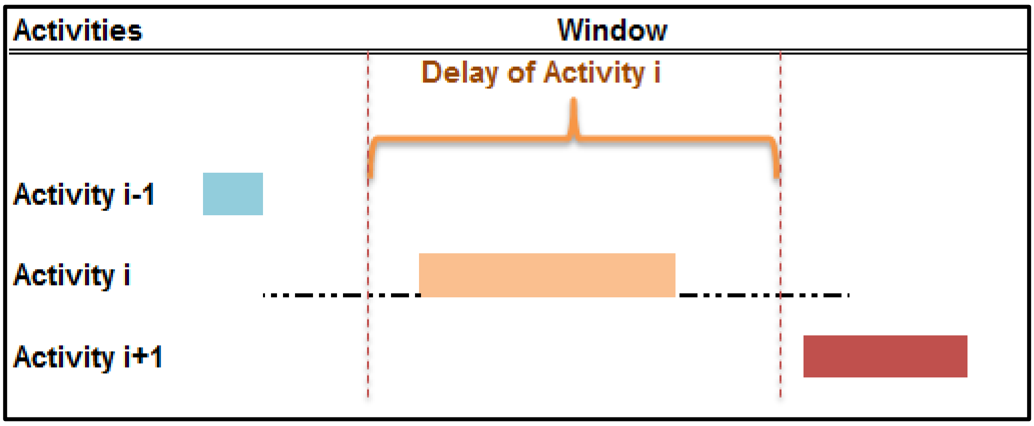


Figure 5.38 Illustration of Formula 5.26 on Gantt Chart (Condition 1)

Additionally, In MS, if ECDs and ENDs don't have any predecessor and ECDs and ENDs commence within one of the previous windows and driven successor activity commences within one of the following windows, then the delays resulting from ECDs and ENDs are computed with Equation (5.18).

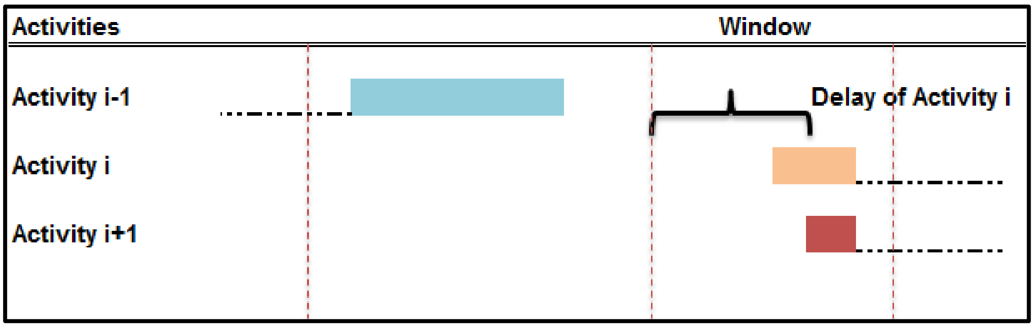


Figure 5.39 Illustration of Formula 5.26 on Gantt Chart (Condition 2)

In MS, if ECDs and ENDs start within one of the previous windows and driven predecessor activity finishes within one of the previous windows and driven successor activity starts within the window, then the delays resulting from ECDs and ENDs are computed with Equation (5.26). Additionally, in MS, if ECDs and ENDs start within one of the previous windows and finish within the window and driven predecessor activity finishes within one of the previous windows and driven successor activity starts within the window, then the delays resulting from ECDs and ENDs are computed with Equation (5.26).

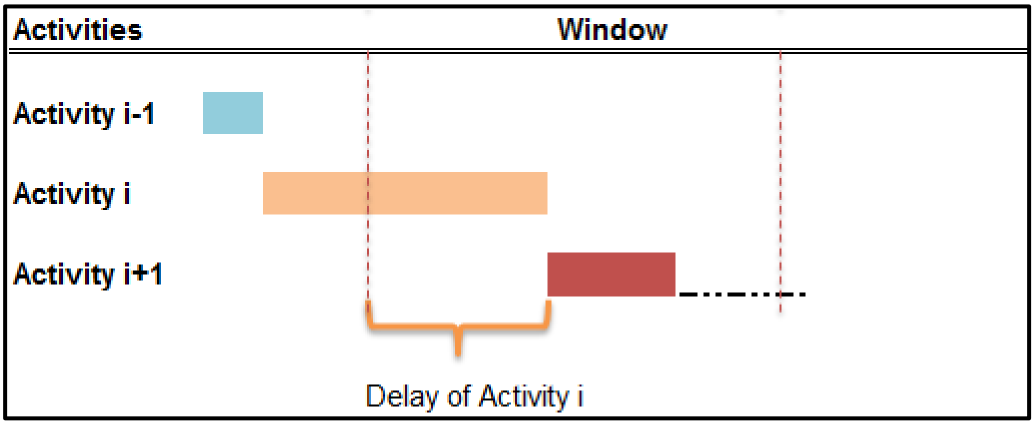


Figure 5.40 Illustration of Formula 5.26 on Gantt Chart (Condition 3)

Moreover, in MS, if ECDs and ENDs start within one of the previous windows and finish within one of the following windows and driven predecessor activity finishes within one of the previous windows and driven successor activity starts within the window, then the delays resulting from ECDs and ENDs are computed with Equation (5.26).

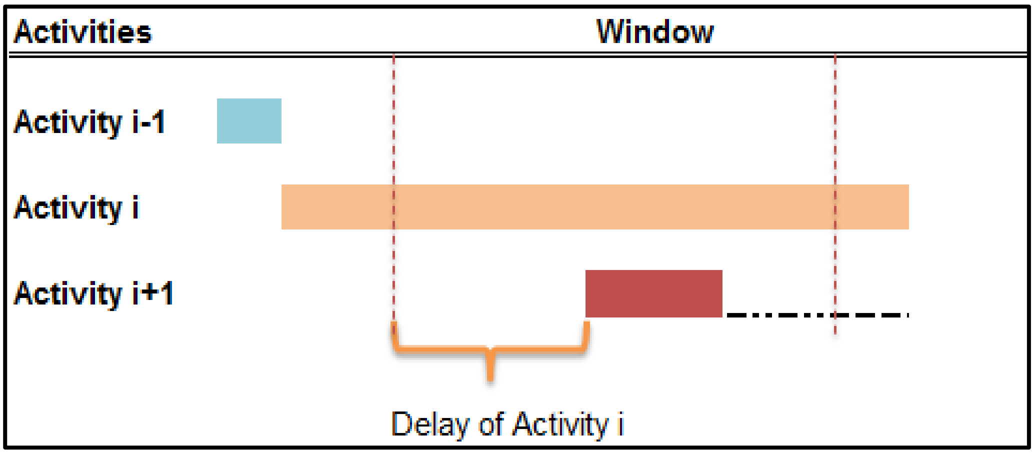


Figure 5.41 Illustration of Formula 5.26 on Gantt Chart (Condition 4)

Furthermore, in MS, if ECDs and ENDs don't have any predecessor and ECDs and ENDs commence within one of the previous windows and driven successor activity commences within the window, then delays resulting from ECDs and ENDs are computed with Equation (5.26).

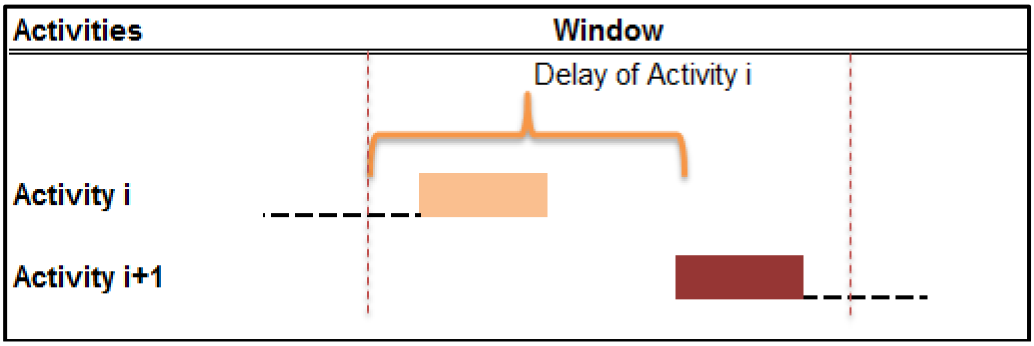


Figure 5.42 Illustration of Formula 5.26 on Gantt Chart

$$\sum_{i=1}^{n}(WCD_i - MC_{D(i+1)}) \quad (5.26)$$

In MS, if ECDs and ENDs start within the window and driven predecessor activity finishes within the window and driven successor activity starts within the window, then the delays resulting from ECDs and ENDs are computed with equation below.

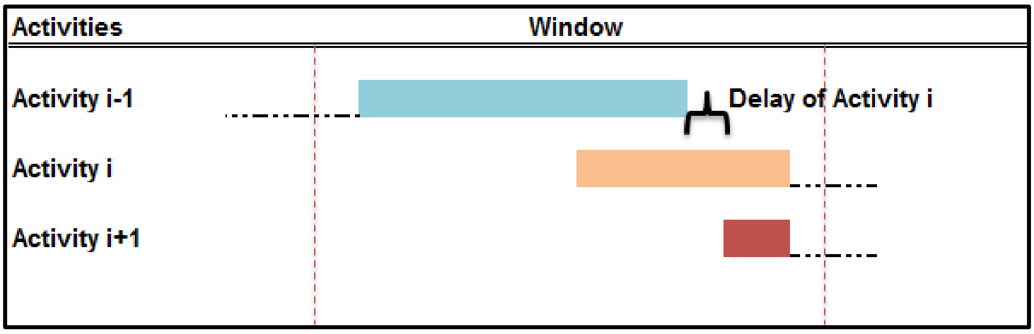


Figure 5.43 Illustration of Formula 5.27 on Gantt Chart

$$\sum_{i=1}^{n} (MF_{D(i-1)} - MC_{D(i+1)}) \tag{5.27}$$

If ECDs and ENDs start within the window in MS and they start after the window in MUS, then no delay is considered.

In MS, if CDs and EDs don't have any successor and CDs and EDs finish within the window and driven predecessor activity finishes within the window, then delay resulting from CDs and EDs are computed as with the Formula (5.28).

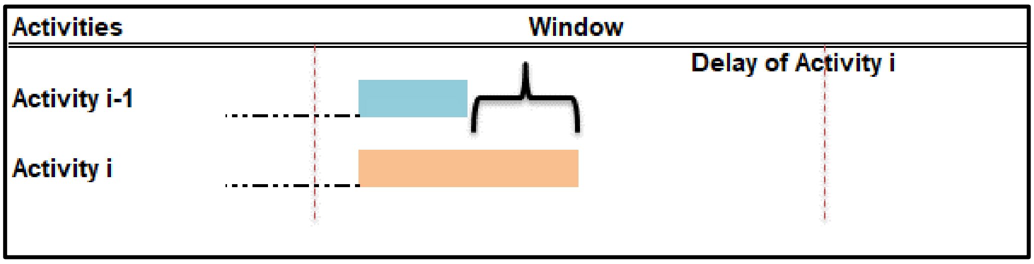


Figure 5.44 Illustration of Formula 5.28 on Gantt Chart

$$\sum_{i=1}^{n} (MF_i - MF_{i-1}) \tag{5.28}$$

In MS, if CDs and EDs don't have any successor and CDs and EDs finish within one of the following windows and driven predecessor activity finishes within the window, then delay resulting from CDs and EDs are computed as with the Formula (5.29).

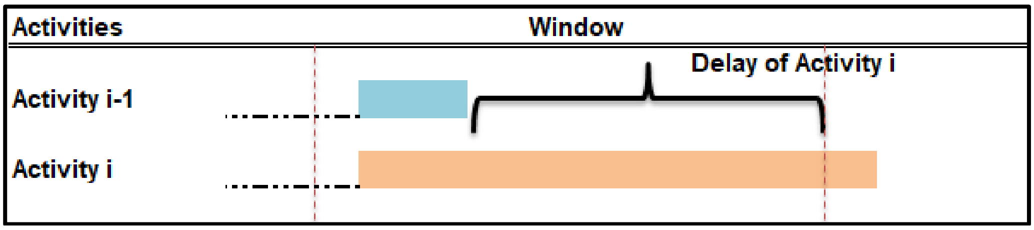


Figure 5.45 Illustration of Formula 5.29 on Gantt Chart

$$\sum_{i=1}^{n}(WFD - MF_{i-1}) \tag{5.29}$$

In MS, if CDs and EDs don't have any successor and CDs and EDs finish within one of the following windows and driven predecessor activity finishes within one of the following windows, then no delay is considered.

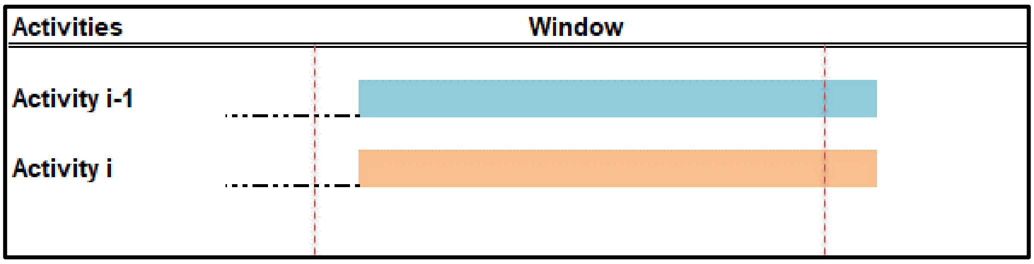


Figure 5.46 Illustration of no delay of CDs and EDs on Gantt Chart (Condition 1)

In MS, if CDs and EDs don't have any successor and CDs and EDs finish within the window and driven predecessor activity finishes within one of the following windows, then no delay is considered.

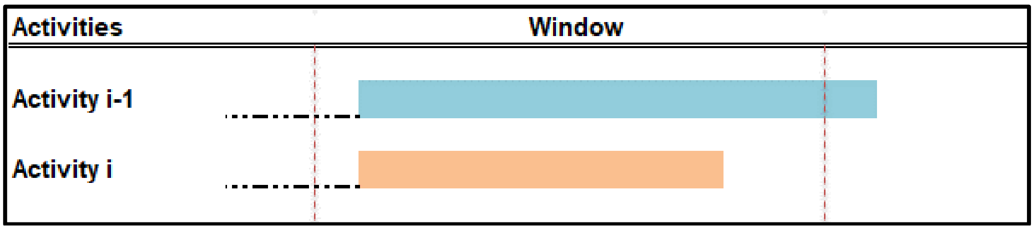


Figure 5.47 Illustration of no delay of CDs and EDs on Gantt Chart (Condition 2)

If CDs and EDs start within the windows in MS and they start after the windows in MUS, then no delay is considered.

If a critical activity has predecessor activity which is added in MUS (which is not in MS) and added activity is in progress in the window then

- Dates (start, finish) of added activity are to be represented in the output table,
- Excusable delays or delay in duration are computed for these added activities according to the type of delay in the concerning window,
- Start or finish date of the added activity is deducted from its successor activity's early and late commencement date by considering the relationships.

5.4 Integration of MSvsMUS into BIM Software as Add-In Tool

At the outset, pseudo codes concerning developed formulations under the section 0 were developed. Later, programming language of Visual Studio (C#) was selected due to the requirements of Bexel Manager's Application Programming Interface (API). Coding is represented under two main headings, namely Developing Core Functionality and User Interface which contains 3 main functions which are Collect User Data, Integrate User Data with Core Functionality and Display Results.

5.4.1 Developing Pseudo Code Concerning Formulations

Pseudo code was developed as depicted under the appendix in order to integrate formulation into Bexel Manager Software.

5.4.2 Developing Core Functionality

This section was developed as per pseudo code provided under the section 5.4.1 via C# programming language.

In order to detect owner-related activities which are added during the schedule updates, a character of '#' is added at the end of the activity; hence, coding will recognize from the character of '#' that the new activity is added due to the owner.

Developed codes are illustrated in Appendix A.

5.4.3 Developing User Interface

This section – which is also developed C# programming language - consists of major two steps, namely 'Collect User Data' and 'Display Result'.

5.4.3.1 Collect User Data

Collect User Data' – which is depicted Figure 5.48 - is designed to be able to enter the information by the user for the windows and to gather data from the 'Developing Core Functionality'. MS, MUS, baseline schedule, window start date and window finish date are required to be entered for each window. When the required data entered through the "Add" button, analysis can be initiated through the "Start Analysis" button. While the "Clear Form" clears the information filled under the area of "Register New Definition", the "Remove" button removes any added window.

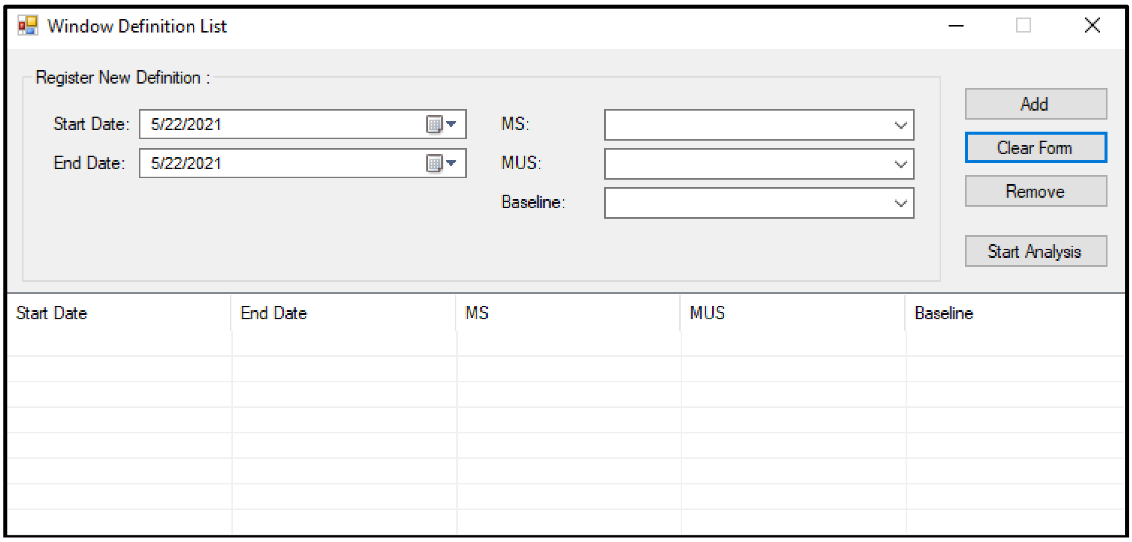


Figure 5.48 User interface area of add-in toll in Bexel Manager

Developed codes are illustrated in Appendix B.

5.4.3.2 Display Results

As is designed in Figure 5.49, 'Display Result' section tabulates the critical activities of each window as an output table including associated equations and their outputs such as delays and improvements.

Results

Activity Name	Windows	∑Delay(Days)	Early or Late Commencement of Activity		Delays or Improvements in Activity Duration		Delays of ECDs and ENDs	
			Equation	Result(Days)	Equation	Result(Days)	Equation	Result(Days)
Total Sum								

Figure 5.49 The design of output table

Developed codes are illustrated in Appendix C.

5.4.4 Installing the Add-in Tool to the Bexel Manager

Generated files in this study, namely MSvsMUS.bxa, MSvsMUS.Controller.dll, MSvsMUS.Model.dll, MSvsMUS.Plugin.dll, MSvsMUS.View.dll are to be copied to the following directory; C:\Users\<user>\\AppData\Roaming\BEXEL\Bexel Manager 21\AddIns. Instead of "<user>", the name of the PC is required to be entered. Following, the developed add-in tool will be shown in the Bexel Manager as illustrated in Figure 5.50.

windows based report

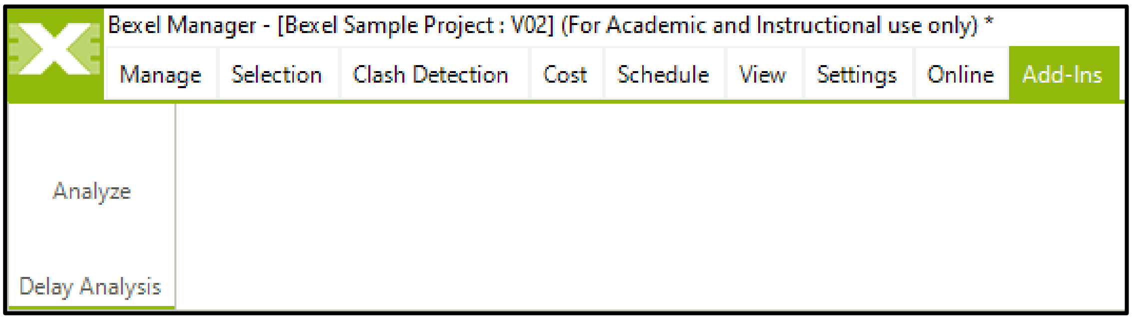


Figure 5.50 Representation of the location of the add-in toll

6
VALIDATIONS

The developed delay analysis method, namely MSvsMUS was validated with a hypothetical study which compared the outcomes of MSvsMUS and Windows Delay Analysis which is the most adopted delay analysis by the courts and practitioners [44]. Next, the developed method was presented via a real airport case study. Later, the developed method and its outcomes obtained from the case study were discussed with 7 experts in an expert panel. Lastly, integration of the developed method into Bexel Manager was validated via a real case study concerning a dwelling project in this section.

6.1 Validation of MSvsMUS

The newly developed method of delay analysis in this thesis is called MSvsMUS. In order to base the reliability of MSvsMUS upon a scientific foundation, the study of validation was first ensured through a hypothetical study in which the outcomes of MSvsMUS were compared with the outcomes Windows Analysis method. Later, a case study was conducted on a real airport project to further validate the method and the data obtained from this case study was presented to 7 experts and discussed in an expert panel as to whether the developed method is efficient and reliable. Each of these three steps was elaborated under three different sections below.

6.1.1 Comparison of MSvsMUS with Windows Analysis: A Hypothetical Study

Although construction projects are unique and delay types vary from project to project [127], [144], occurrence types of delays in project schedules are very much alike. On account of the fact that WA is one of the well-known and accepted delay analysis methods preferred by the construction industry and courts [44], the outputs of MSvsMUS were compared with WA via a case study including 2 windows in order to validate MSvsMUS. Concerning the WA conducted for the case study, Baseline Schedule and the schedule updates associated with the windows are illustrated in the figures below.

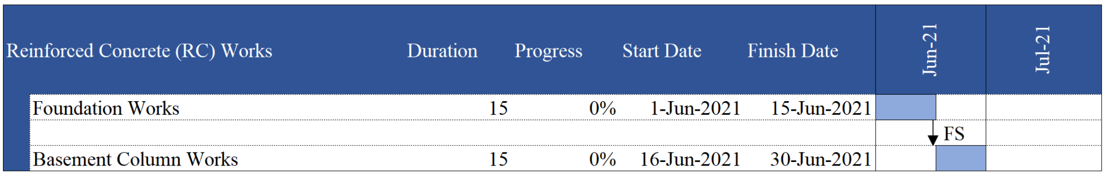


Figure 6.1 Baseline schedule (Cutoff date is 1st June 2021)

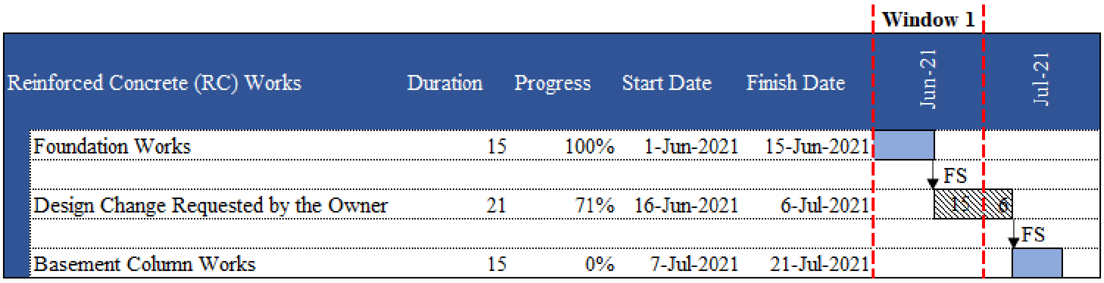


Figure 6.2 Schedule update as of 1st July 2021

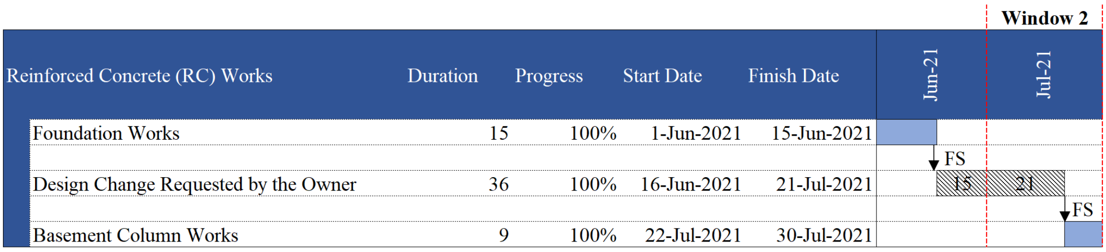


Figure 6.3 Schedule update as of 1st August 2021

Considering the baseline schedule and schedule updates provided above, WA was conducted. As is seen in Window 1 and Window 2, there were 15-day and 21-day delays due to design changes requested by the owner respectively. Furthermore, the contractor improved the "Basement Column Works" for 6 days in Window 2 (Original duration of 15 days – actual duration of 9 days = 6). The results obtained from WA are represented in the table below.

Table 6.1 The outcomes of Windows Analysis

Activity Name	Window #	NEDs		ECDs
		Early or Late Commencement of Activity	Delay in Activity Duration	
Foundation Works	1			
Design Change requested by Owner				-15
Design Change requested by Owner	2			-21
Basement Column Works			6	

Considering MSvsMUS Method, schedules concerning MS and MUS are required. At the outset, the subnet was created concerning the activity of "Design Change requested by the Owner". In the subnet, the planned duration of "Design Change requested by the Owner" is 21 days and it has FS relationship with the predecessor of "Basement Column Works". By following Step 2 provided in the Methodology Section, the foregoing subnet is incorporated into the Baseline Schedule – which is provided in Figure 6.4 - in order to obtain MS for the first window. MS for the first window is illustrated below.

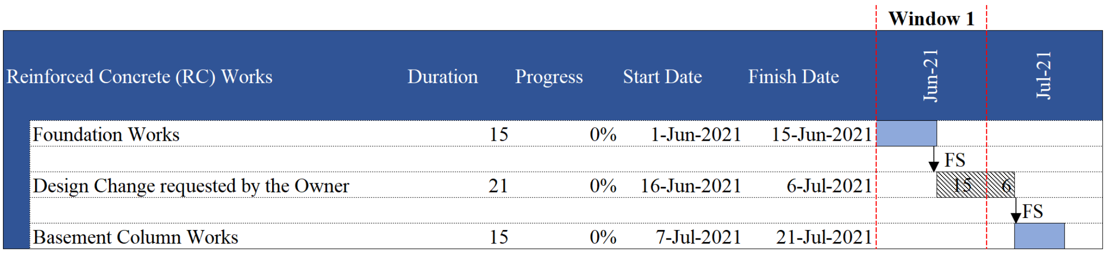


Figure 6.4 Modified schedule for the first window (Cutoff date is 1st June 2021)

According to Step 3, the activities on the critical path for Window 1 are determined, which are "Foundation Work" and "Design Change requested by the Owner". According to Step 4, MS for the first window – which is demonstrated in Figure 8 – is updated to obtain MUS which is provided in Figure 6. MS and MUS are compared as is instructed by Step 5. In Window 1, as is also seen in WA, there is a 15-day delay due to "Design Change requested by the Owner", which is computed via Equation 19 depicted as follow; $MF_{D(i-1)} - WFD_i$ = 15th June 2021 -30th June 2021= -15.

In Window 2, Step 6 and Step 2 are followed respectively to obtain MS for Window 2. According to step 6, MUS of the First Window (see Figure 6.2) is selected, and since there is no new excusable delay in Window 2, MUS of the First Window is considered as MS for the Second Window – which is illustrated in Figure 6.5 – in accordance with Step 2. MS for the second window is illustrated below.

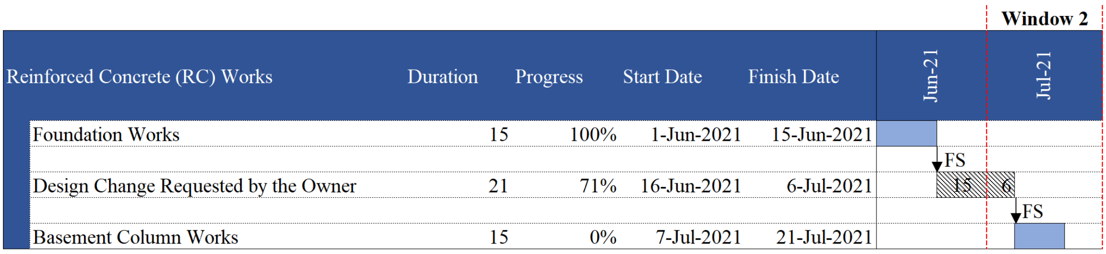


Figure 6.5 Modified schedule for the second window (Cutoff date is 1st July 2021)

By following Step 3, activities on the critical path of the second window – which are "Design Change requested by the Owner" and "Basement Column Works" - are detected. Next, according to Step 4, MS of Window 2 is updated to generate MUS for Window 2. With respect to the comparison of MS with MUS in Window 2, it is clearly identified that "Design Change requested by Owner" delayed for 6 days by the owner, which is computed via Equation 20 as follow; $WSD - MC_{i+1}$ = 1st July 2021 – 7th July 2021 = -6 days. Moreover, "Design Change requested by Owner" delayed by 15 days according to Equation 14 illustrated as follow; $MF_i/\alpha F_{1i} - UF_i/\alpha F_{2i}$ = 6th July 2021 – 21st July 2021 = -15. By the same token, Design Change requested by Owner was planned to be completed within 6 days; however, it was prolonged to 21 days by the contractor, which means the activity delayed for 15 days. "Basement Column Works" was improved for 6 days as computed via Equation 13 as follow; $OD_i/SD_i - ((UF_i/\alpha F_{2i}) - (US_i/\alpha S_{2i}) + 1)$ = 15 – (20th July 2021 – 22nd July 2021 +1) = 6. The summary table involving the results obtained from MSvsMUS is depicted below.

Table 6.2 Outcomes of the MSvsMUS

Activity Name	Window #	NEDs		ECDs
		Early or Late Commencement of Activity	Delay in Activity Duration	
Foundation Works	1			
Design Change requested by Owner				-15
Design Change requested by Owner	2		-15	-6
Basement Column Works			6	

On the whole, when Table 6.1 and Table 6.2 are compared, it can be seen that, unlike WA, the 21-day delay of the "Design Change requested by Owner" in Windows 2 is not only owner-based delay in MSvsMUS. On the contrary, comparing the MS and MUS, MSvsMUS method detected that the 6-day owner's delay was delayed 15 days further by the contractor in Window 2.

6.1.2 Case Study

In order to further validate the developed method, a case study – which has been extracted from a mega airport project containing numerous delays – has been chosen. In the project, the main contractor received 550 site instructions which caused changes in activity sequence and project scope, thus the project delayed for 18 months. The most significant schedule proportion, which contains successive 3 site instructions and the highest number of delays, has been extracted to be examined by this method. This case study, which contains 6 windows, is illustrated in Figure 6.6. Site Instructions such as "Design Revision of Transformer Center", "Design Change of Flyover" and "Design Change of Chiller Plant" are ECDs received by the main contractor and are injected as new activities into the as-built schedules.

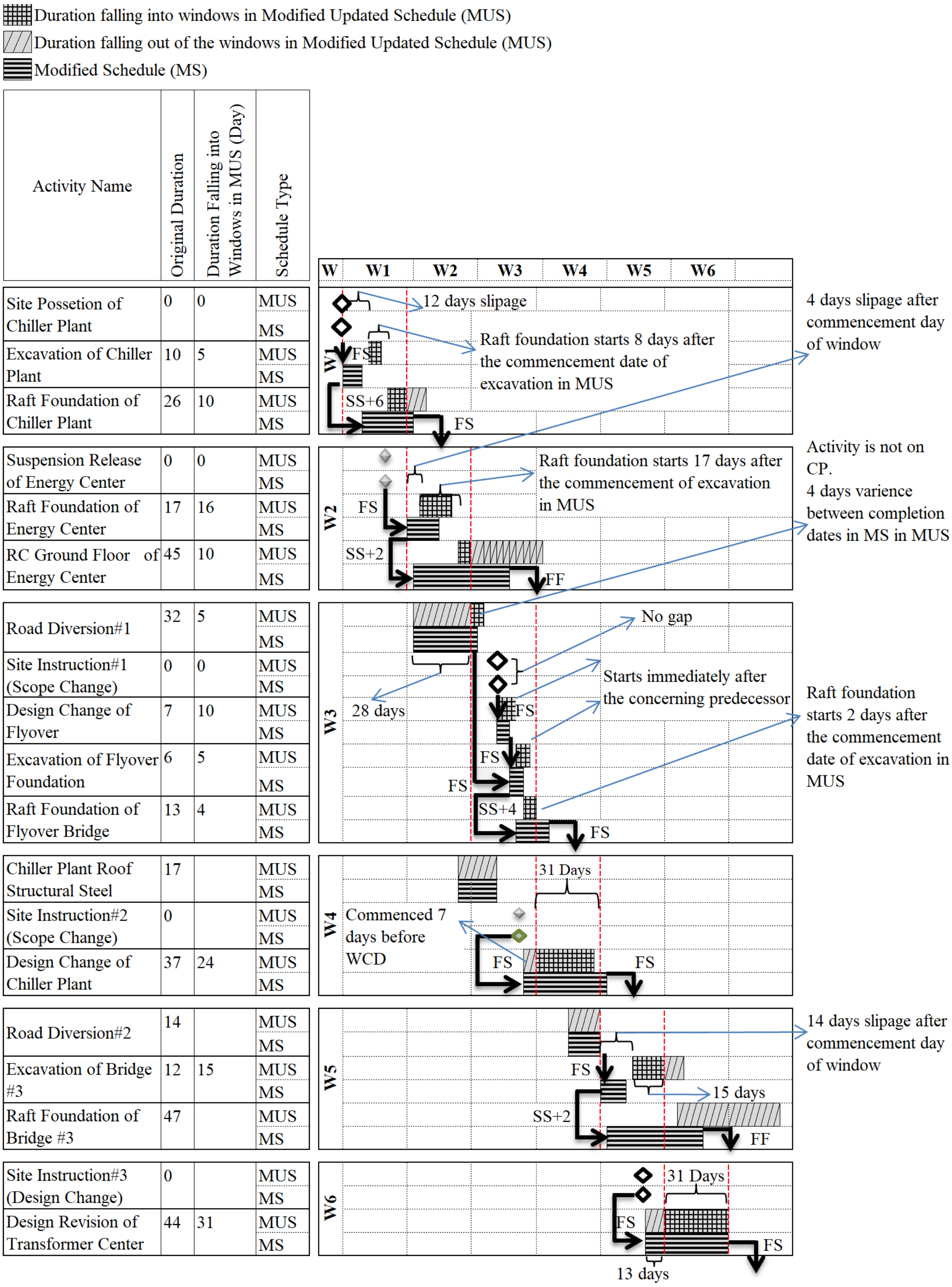


Figure 6.6 Modified schedule vs modified updated schedule of case study

Gantt Chart with explanations of delays and improvements is demonstrated in Figure 6.6. For each activity of 6 windows, original durations of activities and durations of activities falling into windows in MUS are illustrated in Figure 6.6 as well. On Gantt Chart, each window is highlighted with dotted red lines. With the aid of pseudocodes developed in this study, equations and their outcomes for each activity which are on CP are demonstrated in the table below.

Table 6.3 Computation of delays and improvements

Activity Name	Windows	∑Delay (Days)	Early or Late Commencement of Activity		Delays or Improvements in Activity Duration		Delays of ECDs and ENDs	
			Equation	Result (Days)	Equation	Result (Days)	Equation	Result (Days)
Site Possetion of Chiller Plant	1	-14	1	0	13	0	-	0
Excavation of Chiller Plant			3	-12	-	0	-	0
Raft Foundation of Chiller Plant			7	-2	15	0	-	0
Raft Foundation of Energy Center	2	-19	9	-4	-	0	-	0
RC Ground Floor of Energy Center			7	-15	15	0	-	0

Table 6.3 Computation of delays and improvements (continued)

Activity Name	Windows	∑Delay (Days)	Early or Late Commencement of Activity		Delays or Improvements in Activity Duration		Delays of ECDs and ENDs	
			Equation	Result (Days)	Equation	Result (Days)	Equation	Result (Days)
Site Instruction#1 (Scope Change)	3	1	1	0	13	0	18	0
Design Change of Flyover			3	0	13	-3	21	-7
Excavation of Flyover Foundation			3	0	-	0	-	0
Raft Foundation of Flyover Bridge			7	2	13	9	-	0
Design Change of Chiller Plant	4	-24	-	0	6	7	12	-31
Excavation of Bridge #3	5	-27	9	-14	-	0	-	0
Raft Foundation of Bridge #3			11	-13	-	0	-	0
Design Revision of Transformer Center	6	-31	-	0	14	0	12	-31
∑Delay		**-114**		**-58**		**13**		**-69**

∑Delay under Contractor's Responsibility = -45 days

∑Delay under Owner's Responsibility = -69 days

Overall delays under the contractor's responsibility are the summation of the delays resulting from "Early or Late Commencement of Activity" and delays resulting from “Delays or Improvements in Activity Duration”. Overall delays under the owner’s responsibility are represented as “Delays of ECDs and ENDs” in Table 6.3. All in all, as seen in the case study, all the delays are successfully calculated along with their apportioning to the responsible parties. Delays for each window as per contractor responsibility and owner responsibility are clearly demonstrated. Of the total 114-day project delay, a 45-day delay and a 69-day delay are distributed to the contractor’s and the owner’s responsibilities respectively. As is seen inTable 6.3, the relationships among the activities, which are neglected by the existing delay analysis methods, were successfully taken into consideration during the computation of the delays and the improvements resulting from “Early or Late Commencement of Activity". Furthermore, a 3-day further delay and a 7-day improvement were made by the contractor on ECDs, namely Design Change of Flyover and Design Change of Chiller Plant respectively, which is the other novelty of this study.

6.1.3 Evaluation and Discussion of the Developed Delay Analysis Method: An Expert Panel

At the outset of the expert panel, the developed method and its application in the case study were demonstrated to the 7 experts and their opinions were consulted to evaluate the effectiveness and applicability of the developed method. A set of questions that were prepared to elaborate the effectiveness and applicability of MSvsMUS was directed to the participants as is depicted in the methodology section. The findings concerning the strengths and the weaknesses of MSvsMUS derived from the expert panel are demonstrated as follows:

The results of MSvsMUS derived from the case study were evaluated by the experts and the applicability of MSvsMUS in construction projects was confirmed by all of the participants. The experts highlighted that MSvsMUS will be a very important medium through which to provide information concerning delay analysis as per “Early or Late Commencement of Activity”, "Delays or Improvements in Activity Duration” and "Delays of ECDs and ENDs". These details provided through the outcome of this analysis will clear up the disputes in terms of delay or improvements.

Experts concurred that MSvsMUS could successfully compute the actual improvements or further delays made by the contractor on the planned delays of ECDs and ENDs. In the case study concerning a real airport project, the outcome of a further delay of 3-day and a 7-day improvement made by the contractor on ECDs, which are not able to be computed via other delay analysis methods, is one of the novelties of this study. Acceleration cost may also be claimed by the contractor for the improvement made by the contractor on the delay which is under the owner's liability.

Experts agreed on the statement that this method is explicitly much more practical and accurate compared to the other methods since it computes the delays in detail via a defined numerical process considering the relationships of the activities. The experts highlighted in the meeting that considering the relationships of activities provides very accurate outputs. For instance, delay or improvement in the duration of an activity - which has SS relationship type with its driven successor - is not supposed to be computed since concerning delay or improvement doesn't affect the remaining CP. Similarly, in the case study concerning a real airport project, despite having a 5-day improvement, 1-day improvement, 1-day improvement and a 6-day delay in the activities concerning Excavation of Chiller Plant", "Raft Foundation of Energy Center", "Excavation of Flyover Foundation" and "Excavation of Bridge #3" respectively, delays or improvements made on duration were not computed for these activities due to having SS relationship types with their driven successors. Because delays in activity duration can't affect its successor activity in case of existence of SS relationships between the activity and its driven successor activity; therefore, MSvsMUS provides more precise outcomes through the consideration of relationships of FS, FF and SS. Other delay analysis methods are not capable of computing these relationships and this definitely leads to wrong results.

It was also mutually agreed by the experts that consistent applications provided by MSvsMUS can diminish the collisions between the contracting parties thanks to the given procedure and numerical method. In addition, reducing the fluctuations of CP through periodical analysis performed by MSvsMUS was considered a significant advantage by the experts.

Yet, participants met on a common ground that MSvsMUS is quite demanding in terms of project documentation and site records. Performing this developed method in a real project can set consistency; however, it is difficult to manually implement these 21 equations in a large project; therefore, it can be quite challenging. If the software is developed by using these pseudocodes, it could be more practical. On the whole, all the participants asserted that MSvsMUS has the potential to solve the aforementioned problems encountered by the participants in their projects.

6.2 Validation of Add-Inn Tool: A Case Study

To validate the developed add-in tool in Bexel Manager, a case study which is a real dwelling project including 5 windows were generated. Each unit of dwelling was planned to be completed within 134 days with 7 working days per week according to the baseline schedule. 3D views of dwelling drawings were developed via Revit and exported as IFC to be used in Bexel Manager software. Top view (plan) view; back view; front view; right side view; left side view and isometric view of a dwelling are represented as depicted below.

Figure 6.7 Front view of dwelling

Figure 6.8 Back side view of dwelling

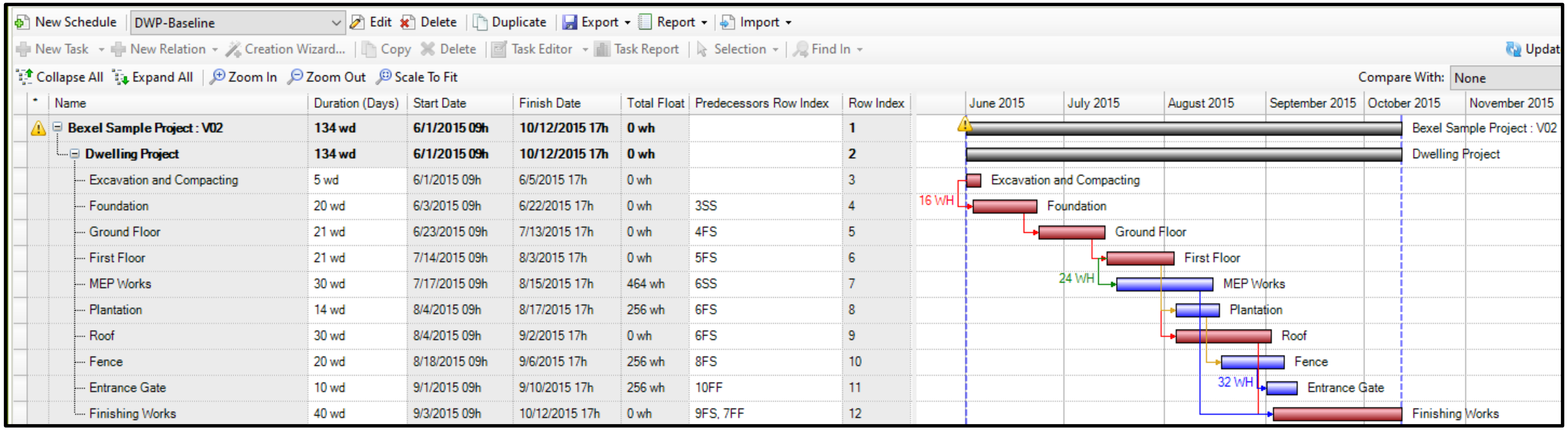
Figure 6.9 Right side view of dwelling

Figure 6.10 Left side view of dwelling

Figure 6.11 Isometric view of dwelling

The baseline schedule, Modified Schedules and Modified Updated Schedules for each window are illustrated as a format of Bexel Manager under this section. Following, the developed add-in tool will be run to compute project delays and improvements as per the owner and contractor. Next, the outputs will be demonstrated and elaborated under this section.

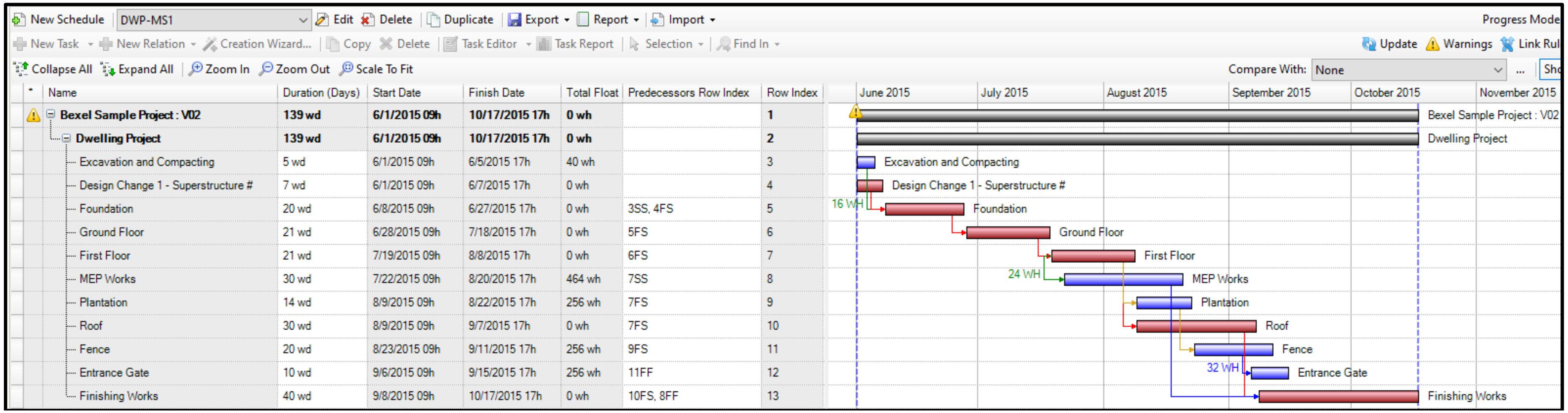
Baseline Schedule is illustrated in Figure 6.12 below.



Name	Duration (Days)	Start Date	Finish Date	Total Float	Predecessors Row Index	Row Index
Bexel Sample Project : V02	**134 wd**	**6/1/2015 09h**	**10/12/2015 17h**	**0 wh**		**1**
Dwelling Project	**134 wd**	**6/1/2015 09h**	**10/12/2015 17h**	**0 wh**		**2**
Excavation and Compacting	5 wd	6/1/2015 09h	6/5/2015 17h	0 wh		3
Foundation	20 wd	6/3/2015 09h	6/22/2015 17h	0 wh	3SS	4
Ground Floor	21 wd	6/23/2015 09h	7/13/2015 17h	0 wh	4FS	5
First Floor	21 wd	7/14/2015 09h	8/3/2015 17h	0 wh	5FS	6
MEP Works	30 wd	7/17/2015 09h	8/15/2015 17h	464 wh	6SS	7
Plantation	14 wd	8/4/2015 09h	8/17/2015 17h	256 wh	6FS	8
Roof	30 wd	8/4/2015 09h	9/2/2015 17h	0 wh	6FS	9
Fence	20 wd	8/18/2015 09h	9/6/2015 17h	256 wh	8FS	10
Entrance Gate	10 wd	9/1/2015 09h	9/10/2015 17h	256 wh	10FF	11
Finishing Works	40 wd	9/3/2015 09h	10/12/2015 17h	0 wh	9FS, 7FF	12

Figure 6.12 Baseline schedule

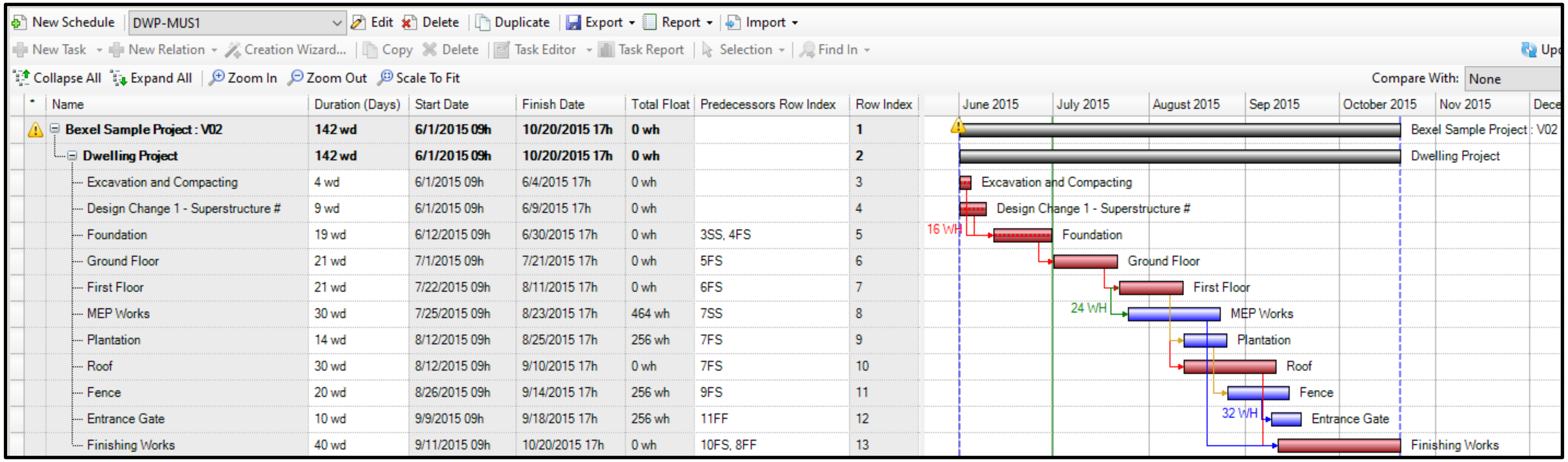
The schedule concerning Modified Schedule for the first window is illustrated in Figure 6.13 below. In Window 1, a new activity named "Design Change 1 #" resulting from the owner's instruction was added.



Name	Duration (Days)	Start Date	Finish Date	Total Float	Predecessors Row Index	Row Index
Bexel Sample Project : V02	**139 wd**	**6/1/2015 09h**	**10/17/2015 17h**	**0 wh**		**1**
Dwelling Project	**139 wd**	**6/1/2015 09h**	**10/17/2015 17h**	**0 wh**		**2**
Excavation and Compacting	5 wd	6/1/2015 09h	6/5/2015 17h	40 wh		3
Design Change 1 - Superstructure #	7 wd	6/1/2015 09h	6/7/2015 17h	0 wh		4
Foundation	20 wd	6/8/2015 09h	6/27/2015 17h	0 wh	3SS, 4FS	5
Ground Floor	21 wd	6/28/2015 09h	7/18/2015 17h	0 wh	5FS	6
First Floor	21 wd	7/19/2015 09h	8/8/2015 17h	0 wh	6FS	7
MEP Works	30 wd	7/22/2015 09h	8/20/2015 17h	464 wh	7SS	8
Plantation	14 wd	8/9/2015 09h	8/22/2015 17h	256 wh	7FS	9
Roof	30 wd	8/9/2015 09h	9/7/2015 17h	0 wh	7FS	10
Fence	20 wd	8/23/2015 09h	9/11/2015 17h	256 wh	9FS	11
Entrance Gate	10 wd	9/6/2015 09h	9/15/2015 17h	256 wh	11FF	12
Finishing Works	40 wd	9/8/2015 09h	10/17/2015 17h	0 wh	10FS, 8FF	13

Figure 6.13 Modified schedule for Window 1

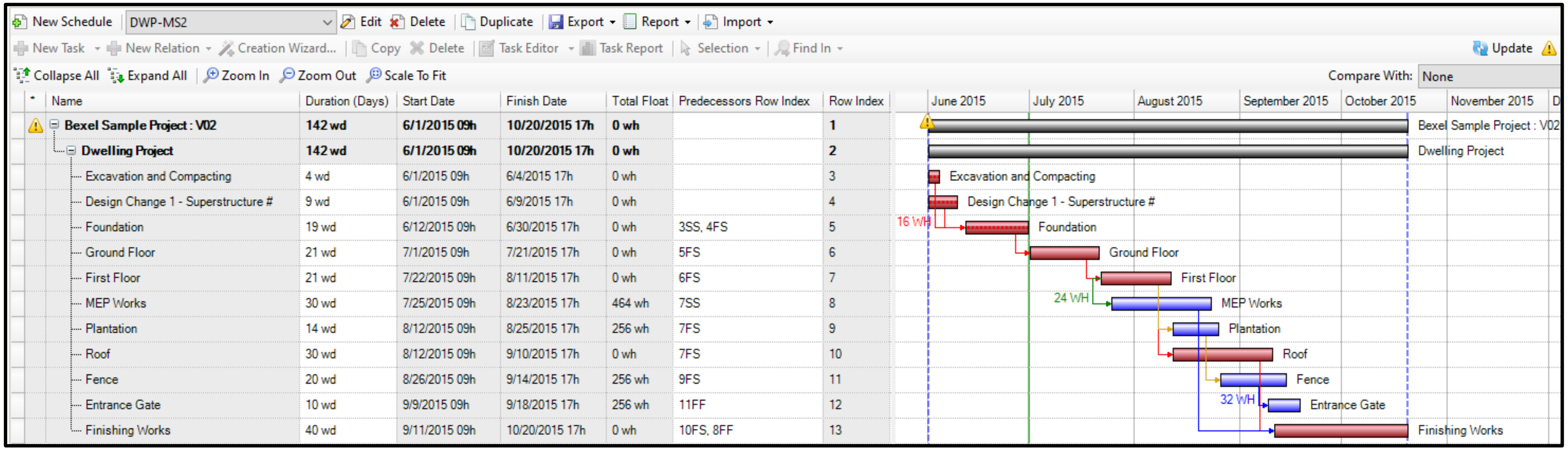
Schedule concerning Modified Updated Schedule for the first window is illustrated in Figure 6.14 below.



Name	Duration (Days)	Start Date	Finish Date	Total Float	Predecessors Row Index	Row Index
Bexel Sample Project : V02	**142 wd**	**6/1/2015 09h**	**10/20/2015 17h**	**0 wh**		**1**
Dwelling Project	**142 wd**	**6/1/2015 09h**	**10/20/2015 17h**	**0 wh**		**2**
Excavation and Compacting	4 wd	6/1/2015 09h	6/4/2015 17h	0 wh		3
Design Change 1 - Superstructure #	9 wd	6/1/2015 09h	6/9/2015 17h	0 wh		4
Foundation	19 wd	6/12/2015 09h	6/30/2015 17h	0 wh	3SS, 4FS	5
Ground Floor	21 wd	7/1/2015 09h	7/21/2015 17h	0 wh	5FS	6
First Floor	21 wd	7/22/2015 09h	8/11/2015 17h	0 wh	6FS	7
MEP Works	30 wd	7/25/2015 09h	8/23/2015 17h	464 wh	7SS	8
Plantation	14 wd	8/12/2015 09h	8/25/2015 17h	256 wh	7FS	9
Roof	30 wd	8/12/2015 09h	9/10/2015 17h	0 wh	7FS	10
Fence	20 wd	8/26/2015 09h	9/14/2015 17h	256 wh	9FS	11
Entrance Gate	10 wd	9/9/2015 09h	9/18/2015 17h	256 wh	11FF	12
Finishing Works	40 wd	9/11/2015 09h	10/20/2015 17h	0 wh	10FS, 8FF	13

Figure 6.14 Modified updated schedule for Window 1

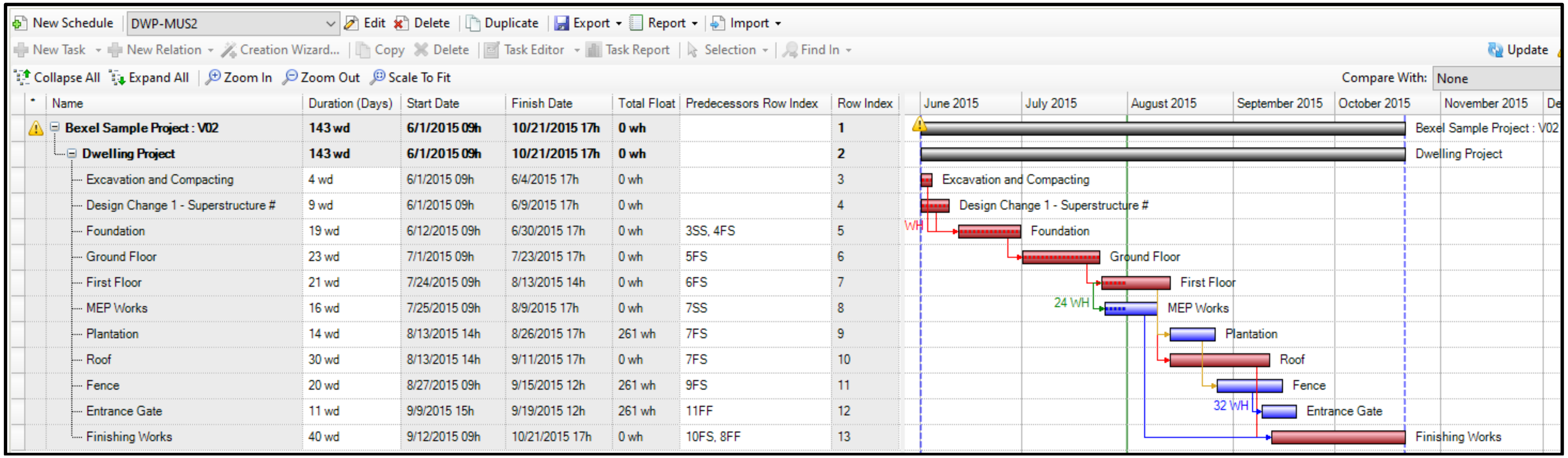
The schedule concerning Modified Schedule for the first window is illustrated in Figure 6.15 below. In Window 2, a new activity named “Design Change 2 #” resulting from the owner’s instruction was added.



Name	Duration (Days)	Start Date	Finish Date	Total Float	Predecessors Row Index	Row Index
Bexel Sample Project : V02	**142 wd**	**6/1/2015 09h**	**10/20/2015 17h**	**0 wh**		**1**
Dwelling Project	**142 wd**	**6/1/2015 09h**	**10/20/2015 17h**	**0 wh**		**2**
Excavation and Compacting	4 wd	6/1/2015 09h	6/4/2015 17h	0 wh		3
Design Change 1 - Superstructure #	9 wd	6/1/2015 09h	6/9/2015 17h	0 wh		4
Foundation	19 wd	6/12/2015 09h	6/30/2015 17h	0 wh	3SS, 4FS	5
Ground Floor	21 wd	7/1/2015 09h	7/21/2015 17h	0 wh	5FS	6
First Floor	21 wd	7/22/2015 09h	8/11/2015 17h	0 wh	6FS	7
MEP Works	30 wd	7/25/2015 09h	8/23/2015 17h	464 wh	7SS	8
Plantation	14 wd	8/12/2015 09h	8/25/2015 17h	256 wh	7FS	9
Roof	30 wd	8/12/2015 09h	9/10/2015 17h	0 wh	7FS	10
Fence	20 wd	8/26/2015 09h	9/14/2015 17h	256 wh	9FS	11
Entrance Gate	10 wd	9/9/2015 09h	9/18/2015 17h	256 wh	11FF	12
Finishing Works	40 wd	9/11/2015 09h	10/20/2015 17h	0 wh	10FS, 8FF	13

Figure 6.15 Modified schedule for Window 2

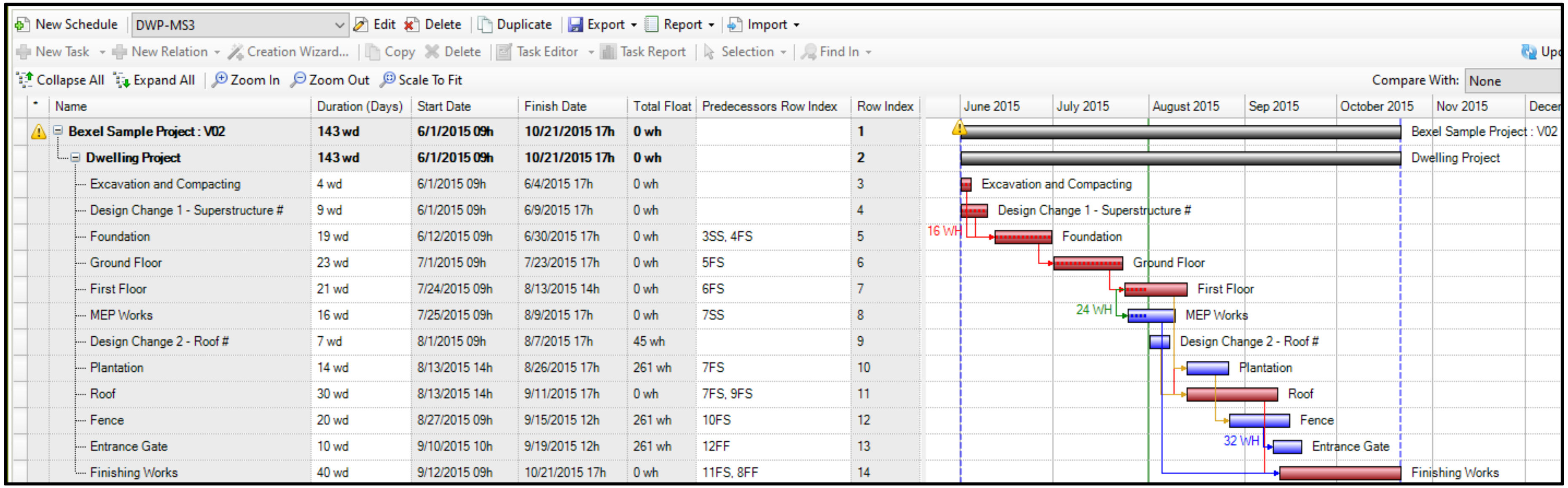
The schedule concerning Modified Updated Schedule for the first window is illustrated in Figure 6.16 below



Name	Duration (Days)	Start Date	Finish Date	Total Float	Predecessors Row Index	Row Index
Bexel Sample Project : V02	**143 wd**	**6/1/2015 09h**	**10/21/2015 17h**	**0 wh**		**1**
Dwelling Project	**143 wd**	**6/1/2015 09h**	**10/21/2015 17h**	**0 wh**		**2**
Excavation and Compacting	4 wd	6/1/2015 09h	6/4/2015 17h	0 wh		3
Design Change 1 - Superstructure #	9 wd	6/1/2015 09h	6/9/2015 17h	0 wh		4
Foundation	19 wd	6/12/2015 09h	6/30/2015 17h	0 wh	3SS, 4FS	5
Ground Floor	23 wd	7/1/2015 09h	7/23/2015 17h	0 wh	5FS	6
First Floor	21 wd	7/24/2015 09h	8/13/2015 14h	0 wh	6FS	7
MEP Works	16 wd	7/25/2015 09h	8/9/2015 17h	0 wh	7SS	8
Plantation	14 wd	8/13/2015 14h	8/26/2015 17h	261 wh	7FS	9
Roof	30 wd	8/13/2015 14h	9/11/2015 17h	0 wh	7FS	10
Fence	20 wd	8/27/2015 09h	9/15/2015 12h	261 wh	9FS	11
Entrance Gate	11 wd	9/9/2015 15h	9/19/2015 12h	261 wh	11FF	12
Finishing Works	40 wd	9/12/2015 09h	10/21/2015 17h	0 wh	10FS, 8FF	13

Figure 6.16 Modified updated schedule for Window 2

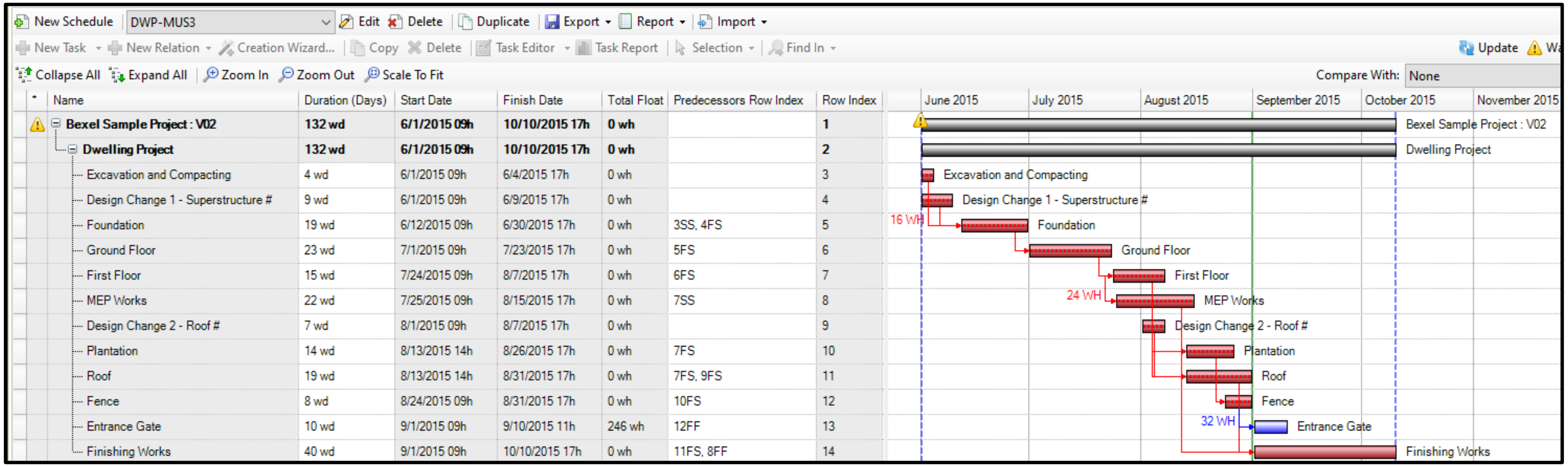
The schedule concerning Modified Schedule for the first window is illustrated in Figure 6.17 below.



Name	Duration (Days)	Start Date	Finish Date	Total Float	Predecessors Row Index	Row Index
Bexel Sample Project : V02	**143 wd**	**6/1/2015 09h**	**10/21/2015 17h**	**0 wh**		**1**
Dwelling Project	**143 wd**	**6/1/2015 09h**	**10/21/2015 17h**	**0 wh**		**2**
Excavation and Compacting	4 wd	6/1/2015 09h	6/4/2015 17h	0 wh		3
Design Change 1 - Superstructure #	9 wd	6/1/2015 09h	6/9/2015 17h	0 wh		4
Foundation	19 wd	6/12/2015 09h	6/30/2015 17h	0 wh	3SS, 4FS	5
Ground Floor	23 wd	7/1/2015 09h	7/23/2015 17h	0 wh	5FS	6
First Floor	21 wd	7/24/2015 09h	8/13/2015 14h	0 wh	6FS	7
MEP Works	16 wd	7/25/2015 09h	8/9/2015 17h	0 wh	7SS	8
Design Change 2 - Roof #	7 wd	8/1/2015 09h	8/7/2015 17h	45 wh		9
Plantation	14 wd	8/13/2015 14h	8/26/2015 17h	261 wh	7FS	10
Roof	30 wd	8/13/2015 14h	9/11/2015 17h	0 wh	7FS, 9FS	11
Fence	20 wd	8/27/2015 09h	9/15/2015 12h	261 wh	10FS	12
Entrance Gate	10 wd	9/10/2015 10h	9/19/2015 12h	261 wh	12FF	13
Finishing Works	40 wd	9/12/2015 09h	10/21/2015 17h	0 wh	11FS, 8FF	14

Figure 6.17 Modified schedule for Window 3

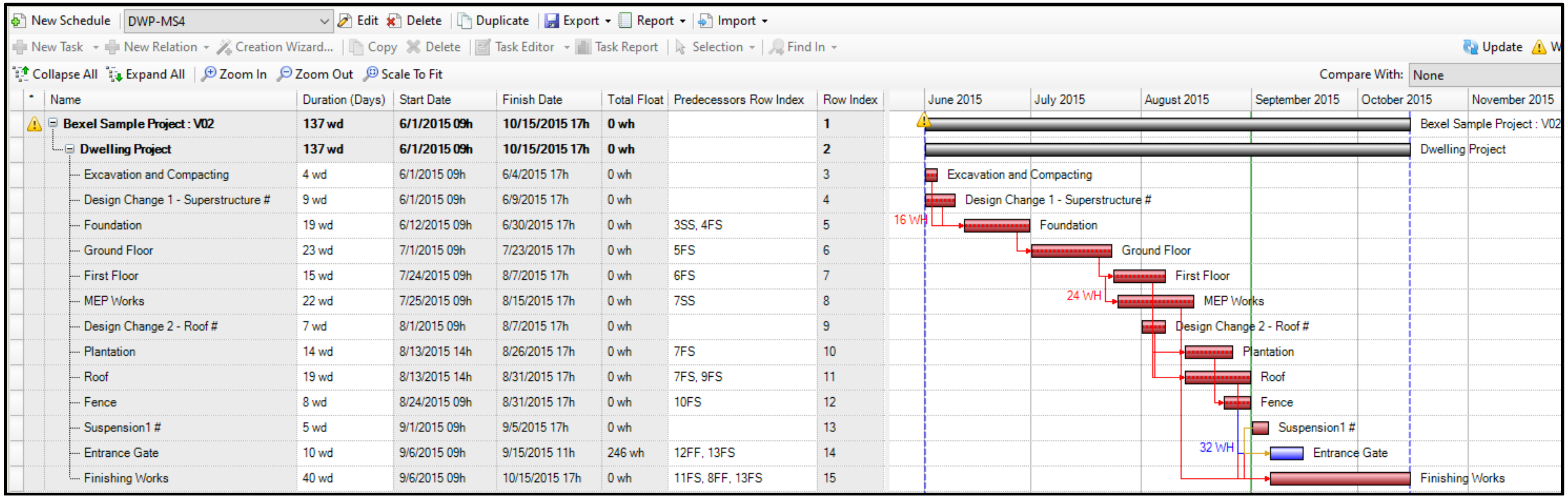
The schedule concerning Modified Updated Schedule for the first window is illustrated in Figure 6.18 below.



Name	Duration (Days)	Start Date	Finish Date	Total Float	Predecessors Row Index	Row Index
Bexel Sample Project : V02	**132 wd**	**6/1/2015 09h**	**10/10/2015 17h**	**0 wh**		**1**
Dwelling Project	**132 wd**	**6/1/2015 09h**	**10/10/2015 17h**	**0 wh**		**2**
Excavation and Compacting	4 wd	6/1/2015 09h	6/4/2015 17h	0 wh		3
Design Change 1 - Superstructure #	9 wd	6/1/2015 09h	6/9/2015 17h	0 wh		4
Foundation	19 wd	6/12/2015 09h	6/30/2015 17h	0 wh	3SS, 4FS	5
Ground Floor	23 wd	7/1/2015 09h	7/23/2015 17h	0 wh	5FS	6
First Floor	15 wd	7/24/2015 09h	8/7/2015 17h	0 wh	6FS	7
MEP Works	22 wd	7/25/2015 09h	8/15/2015 17h	0 wh	7SS	8
Design Change 2 - Roof #	7 wd	8/1/2015 09h	8/7/2015 17h	0 wh		9
Plantation	14 wd	8/13/2015 14h	8/26/2015 17h	0 wh	7FS	10
Roof	19 wd	8/13/2015 14h	8/31/2015 17h	0 wh	7FS, 9FS	11
Fence	8 wd	8/24/2015 09h	8/31/2015 17h	0 wh	10FS	12
Entrance Gate	10 wd	9/1/2015 09h	9/10/2015 11h	246 wh	12FF	13
Finishing Works	40 wd	9/1/2015 09h	10/10/2015 17h	0 wh	11FS, 8FF	14

Figure 6.18 Modified updated schedule for Window 3

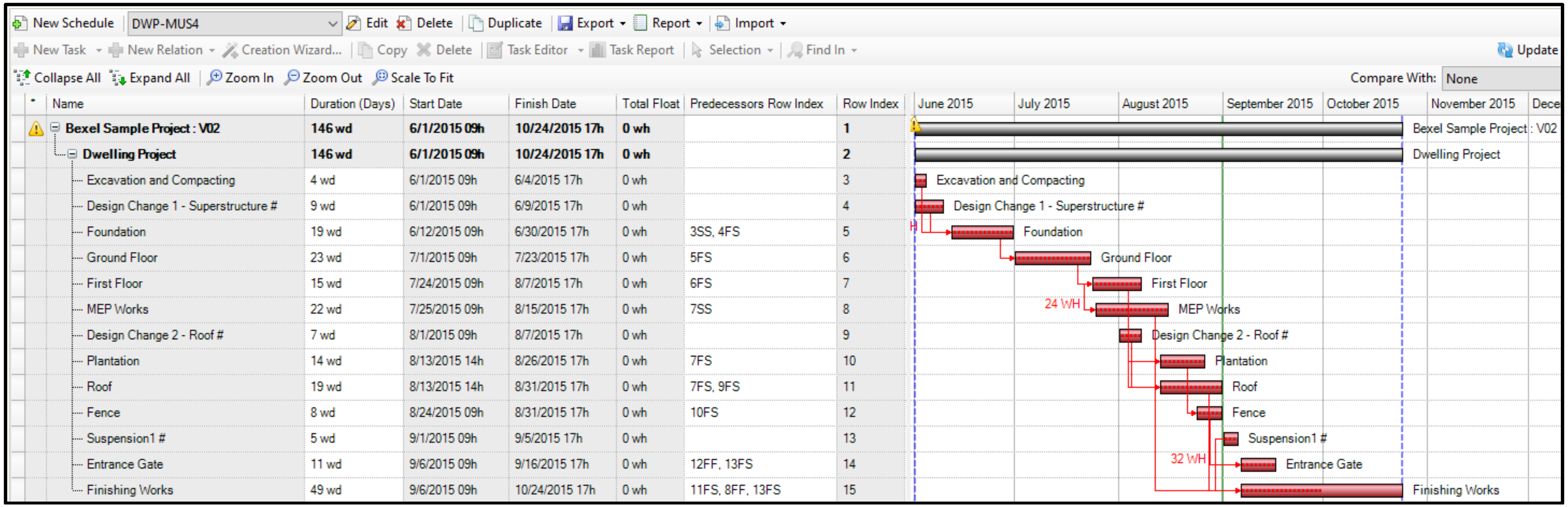
The schedule concerning Modified Schedule for the first window is illustrated in Figure 6.19 below.



Name	Duration (Days)	Start Date	Finish Date	Total Float	Predecessors Row Index	Row Index
Bexel Sample Project : V02	**137 wd**	**6/1/2015 09h**	**10/15/2015 17h**	**0 wh**		**1**
Dwelling Project	**137 wd**	**6/1/2015 09h**	**10/15/2015 17h**	**0 wh**		**2**
Excavation and Compacting	4 wd	6/1/2015 09h	6/4/2015 17h	0 wh		3
Design Change 1 - Superstructure #	9 wd	6/1/2015 09h	6/9/2015 17h	0 wh		4
Foundation	19 wd	6/12/2015 09h	6/30/2015 17h	0 wh	3SS, 4FS	5
Ground Floor	23 wd	7/1/2015 09h	7/23/2015 17h	0 wh	5FS	6
First Floor	15 wd	7/24/2015 09h	8/7/2015 17h	0 wh	6FS	7
MEP Works	22 wd	7/25/2015 09h	8/15/2015 17h	0 wh	7SS	8
Design Change 2 - Roof #	7 wd	8/1/2015 09h	8/7/2015 17h	0 wh		9
Plantation	14 wd	8/13/2015 14h	8/26/2015 17h	0 wh	7FS	10
Roof	19 wd	8/13/2015 14h	8/31/2015 17h	0 wh	7FS, 9FS	11
Fence	8 wd	8/24/2015 09h	8/31/2015 17h	0 wh	10FS	12
Suspension1 #	5 wd	9/1/2015 09h	9/5/2015 17h	0 wh		13
Entrance Gate	10 wd	9/6/2015 09h	9/15/2015 11h	246 wh	12FF, 13FS	14
Finishing Works	40 wd	9/6/2015 09h	10/15/2015 17h	0 wh	11FS, 8FF, 13FS	15

Figure 6.19 Modified schedule for Window 4

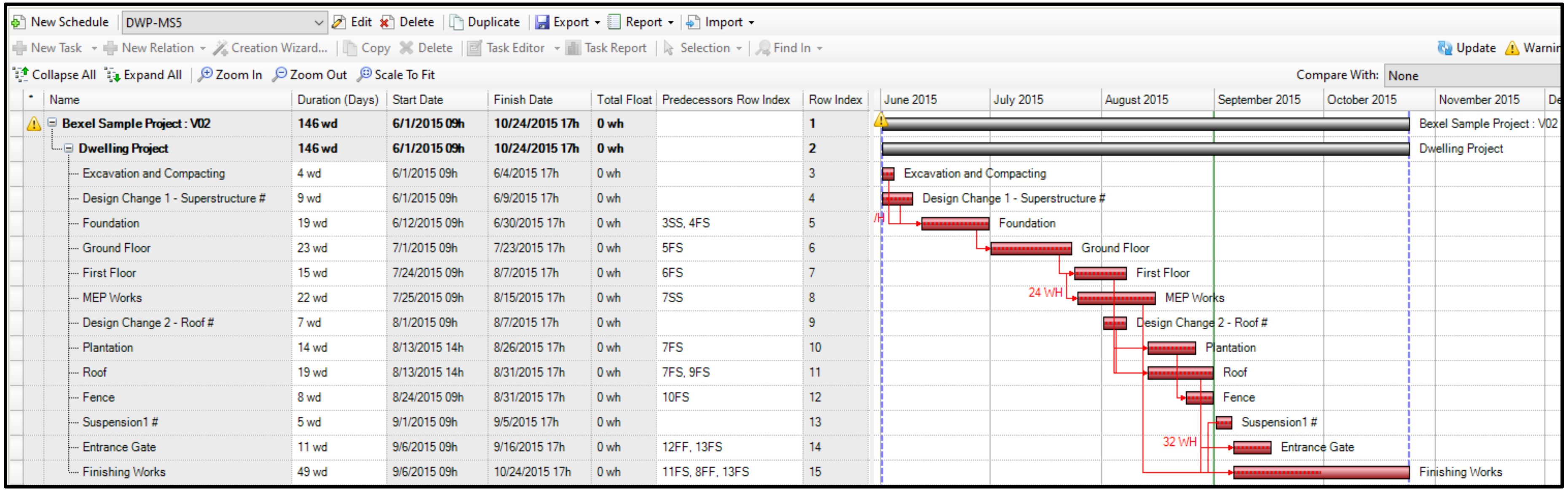
The schedule concerning Modified Updated Schedule for the first window is illustrated in Figure 6.20 below.



Name	Duration (Days)	Start Date	Finish Date	Total Float	Predecessors Row Index	Row Index
Bexel Sample Project : V02	**146 wd**	**6/1/2015 09h**	**10/24/2015 17h**	**0 wh**		**1**
Dwelling Project	**146 wd**	**6/1/2015 09h**	**10/24/2015 17h**	**0 wh**		**2**
Excavation and Compacting	4 wd	6/1/2015 09h	6/4/2015 17h	0 wh		3
Design Change 1 - Superstructure #	9 wd	6/1/2015 09h	6/9/2015 17h	0 wh		4
Foundation	19 wd	6/12/2015 09h	6/30/2015 17h	0 wh	3SS, 4FS	5
Ground Floor	23 wd	7/1/2015 09h	7/23/2015 17h	0 wh	5FS	6
First Floor	15 wd	7/24/2015 09h	8/7/2015 17h	0 wh	6FS	7
MEP Works	22 wd	7/25/2015 09h	8/15/2015 17h	0 wh	7SS	8
Design Change 2 - Roof #	7 wd	8/1/2015 09h	8/7/2015 17h	0 wh		9
Plantation	14 wd	8/13/2015 14h	8/26/2015 17h	0 wh	7FS	10
Roof	19 wd	8/13/2015 14h	8/31/2015 17h	0 wh	7FS, 9FS	11
Fence	8 wd	8/24/2015 09h	8/31/2015 17h	0 wh	10FS	12
Suspension1 #	5 wd	9/1/2015 09h	9/5/2015 17h	0 wh		13
Entrance Gate	11 wd	9/6/2015 09h	9/16/2015 17h	0 wh	12FF, 13FS	14
Finishing Works	49 wd	9/6/2015 09h	10/24/2015 17h	0 wh	11FS, 8FF, 13FS	15

Figure 6.20 Modified updated schedule for Window 4

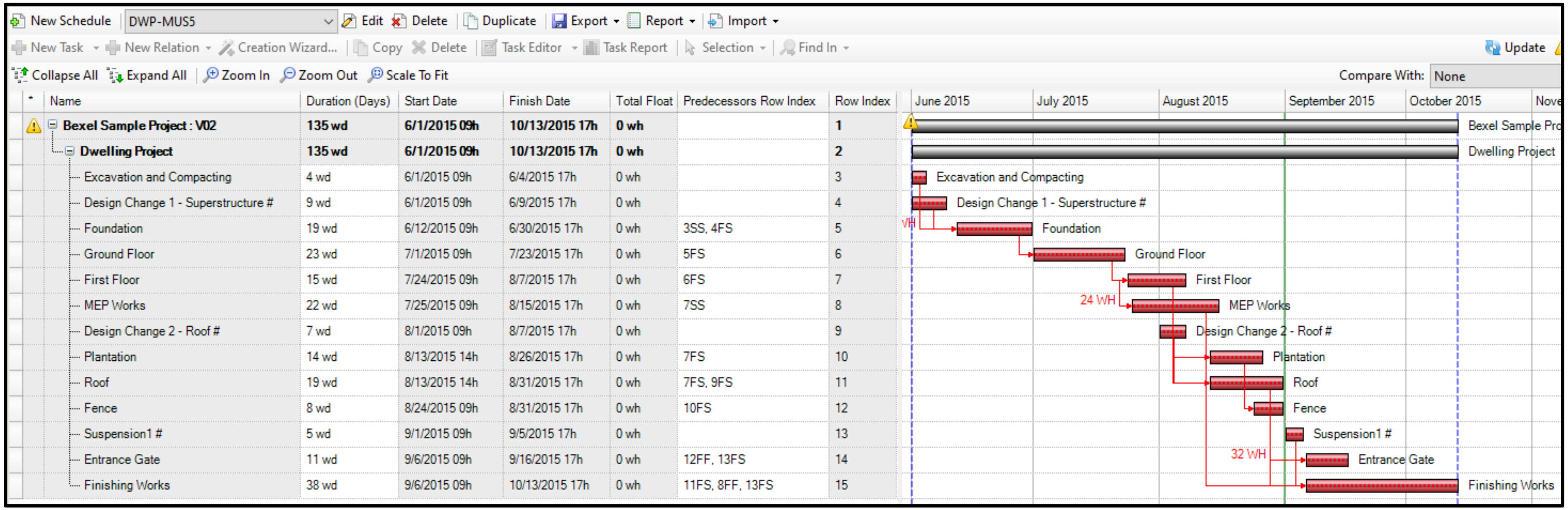
The schedule concerning Modified Schedule for the first window is illustrated in Figure 6.21 below. In Window 5, a new activity named "Suspension #" resulting from the owner's instruction was added.



Name	Duration (Days)	Start Date	Finish Date	Total Float	Predecessors Row Index	Row Index
Bexel Sample Project : V02	**146 wd**	**6/1/2015 09h**	**10/24/2015 17h**	**0 wh**		**1**
Dwelling Project	**146 wd**	**6/1/2015 09h**	**10/24/2015 17h**	**0 wh**		**2**
Excavation and Compacting	4 wd	6/1/2015 09h	6/4/2015 17h	0 wh		3
Design Change 1 - Superstructure #	9 wd	6/1/2015 09h	6/9/2015 17h	0 wh		4
Foundation	19 wd	6/12/2015 09h	6/30/2015 17h	0 wh	3SS, 4FS	5
Ground Floor	23 wd	7/1/2015 09h	7/23/2015 17h	0 wh	5FS	6
First Floor	15 wd	7/24/2015 09h	8/7/2015 17h	0 wh	6FS	7
MEP Works	22 wd	7/25/2015 09h	8/15/2015 17h	0 wh	7SS	8
Design Change 2 - Roof #	7 wd	8/1/2015 09h	8/7/2015 17h	0 wh		9
Plantation	14 wd	8/13/2015 14h	8/26/2015 17h	0 wh	7FS	10
Roof	19 wd	8/13/2015 14h	8/31/2015 17h	0 wh	7FS, 9FS	11
Fence	8 wd	8/24/2015 09h	8/31/2015 17h	0 wh	10FS	12
Suspension1 #	5 wd	9/1/2015 09h	9/5/2015 17h	0 wh		13
Entrance Gate	11 wd	9/6/2015 09h	9/16/2015 17h	0 wh	12FF, 13FS	14
Finishing Works	49 wd	9/6/2015 09h	10/24/2015 17h	0 wh	11FS, 8FF, 13FS	15

Figure 6.21 Modified schedule for Window 5

The schedule concerning Modified Updated Schedule for the first window is illustrated in Figure 6.22 below.



Name	Duration (Days)	Start Date	Finish Date	Total Float	Predecessors Row Index	Row Index
Bexel Sample Project : V02	**135 wd**	**6/1/2015 09h**	**10/13/2015 17h**	**0 wh**		**1**
Dwelling Project	**135 wd**	**6/1/2015 09h**	**10/13/2015 17h**	**0 wh**		**2**
Excavation and Compacting	4 wd	6/1/2015 09h	6/4/2015 17h	0 wh		3
Design Change 1 - Superstructure #	9 wd	6/1/2015 09h	6/9/2015 17h	0 wh		4
Foundation	19 wd	6/12/2015 09h	6/30/2015 17h	0 wh	3SS, 4FS	5
Ground Floor	23 wd	7/1/2015 09h	7/23/2015 17h	0 wh	5FS	6
First Floor	15 wd	7/24/2015 09h	8/7/2015 17h	0 wh	6FS	7
MEP Works	22 wd	7/25/2015 09h	8/15/2015 17h	0 wh	7SS	8
Design Change 2 - Roof #	7 wd	8/1/2015 09h	8/7/2015 17h	0 wh		9
Plantation	14 wd	8/13/2015 14h	8/26/2015 17h	0 wh	7FS	10
Roof	19 wd	8/13/2015 14h	8/31/2015 17h	0 wh	7FS, 9FS	11
Fence	8 wd	8/24/2015 09h	8/31/2015 17h	0 wh	10FS	12
Suspension1 #	5 wd	9/1/2015 09h	9/5/2015 17h	0 wh		13
Entrance Gate	11 wd	9/6/2015 09h	9/16/2015 17h	0 wh	12FF, 13FS	14
Finishing Works	38 wd	9/6/2015 09h	10/13/2015 17h	0 wh	11FS, 8FF, 13FS	15

Figure 6.22 Modified updated schedule for Window 5

Information related to each window obtained from the baseline schedule, modified schedules and modified updated schedule is entered into the user interface area as depicted in Figure 6.23 below.

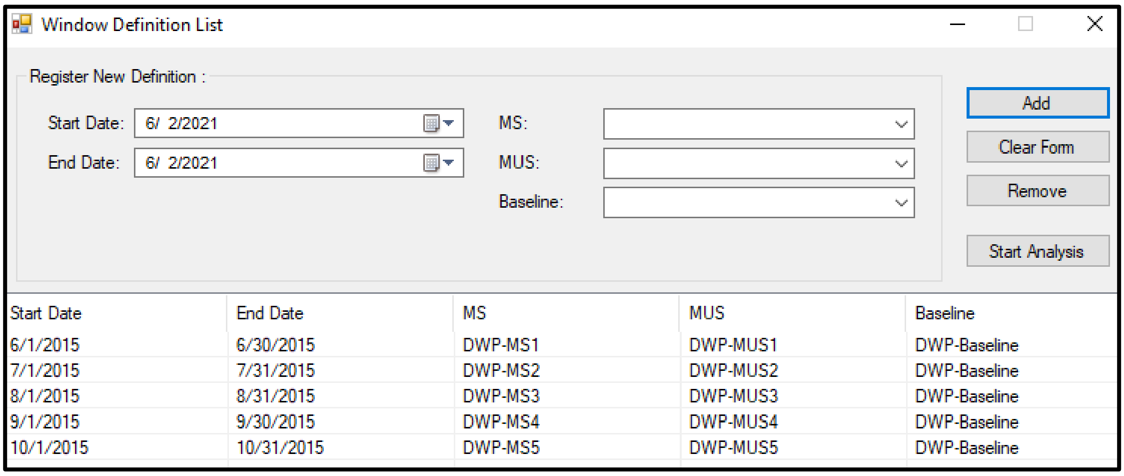


Figure 6.23 Information of each window under user interface area

In the wake of analysis, the user interface area shows up as illustrated below.

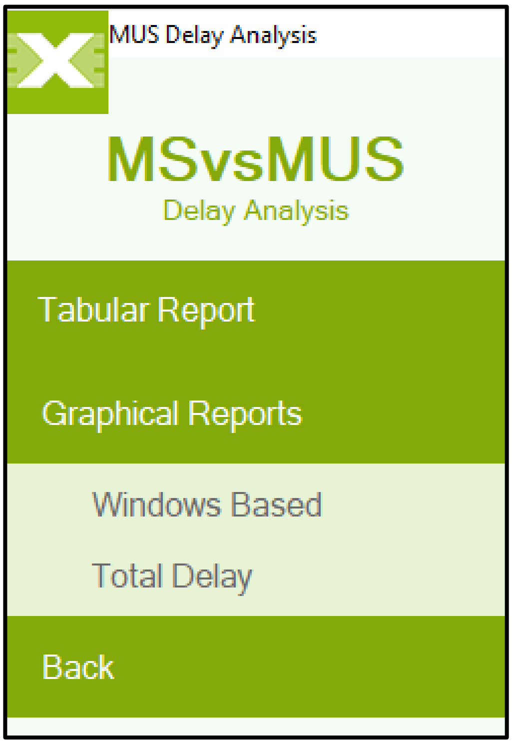


Figure 6.24 Illustration of user interface concerning report section

Tabular report provides all delayed and improved activities as per the number of windows, “Early and Late Commencement of Activity”, “Delays or Improvement in Activity Duration” and “Delays of ECDs and ENDs”. Additionally, Activity Based report provides total delays and improvements as per activity name. Window Based report provides total delay and improvement report as per windows in a pie chart. Total Delay reports deliver total delays and improvements in bar chart. The output table obtained from the MSvsMUS is represented in Figure 6.25 below.

Tabular Report								
Activity Name	Window	∑Delay(Days)	Non-Excusable Delays				Excusable Delays	
			Early or Late Commencement of Activity		Delays or Improvements in Activity Duration		Delays of ECDs and ENDs	
			Result(Days)		Result(Days)		Result(Days)	
Ground Floor	1	-11	10	0	-		-	
Foundation			3	-3	13	1	-	
Design Change 1 - Superstructure #			1	0	13	-2	18	-7
Ground Floor	2	-2	9	0	13	-2	-	
First Floor			3	0	-		-	
Roof	3	6	3	-5	13	11	-	
Finishing Works	4	-5	3	0	15	0	-	
Suspension1 #			1	0	13	0	18	-5
Finishing Works	5	11	-		14	11	-	
Total Sum		**-1**	**-8**		**19**		**-12**	

Figure 6.25 Output table obtained from MSvsMUS add-in tool of Bexel Manager

Activity Name	Overall Delay or Improvement	Early and Late Commencement of Activity	Delays or Improvement in Activity Duration	Delays of ECDs and ENDs
Design Change 1 - Superstructure #	-9		-2	-7
Foundation	-2	-3	1	
Ground Floor	-2		-2	
First Floor	0	0		
Roof	6	-5	11	
Suspension1 #	-5			-5
Finishing Works	11		11	
Subtotal	**-1**	**-8**	**19**	**-12**

Figure 6.26 Total delays and improvements as per activity name obtained from MSvsMUS add-in tool of Bexel Manager

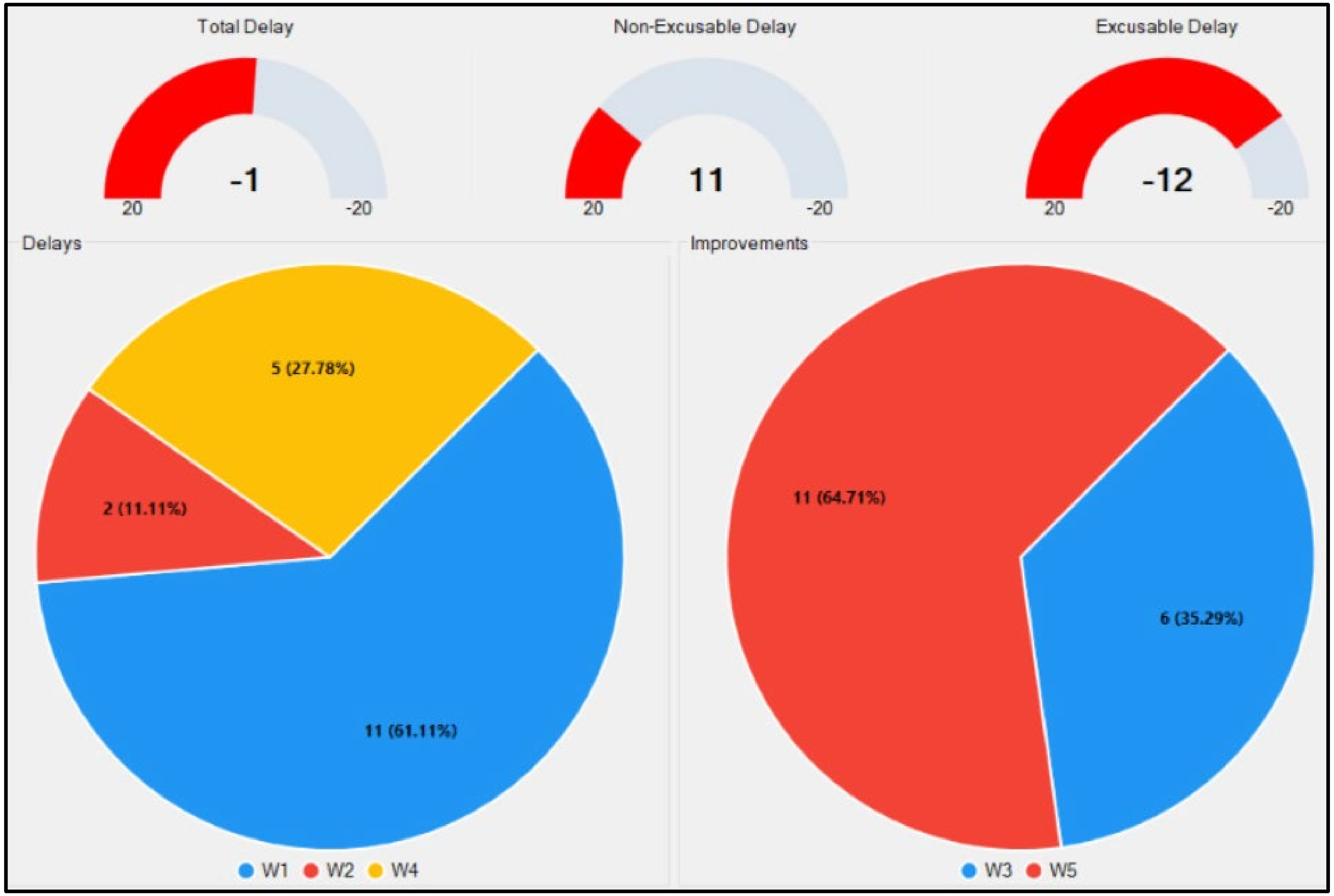


Figure 6.27 Total delay and improvement report as per windows obtained from MSvsMUS add-in tool of Bexel Manager

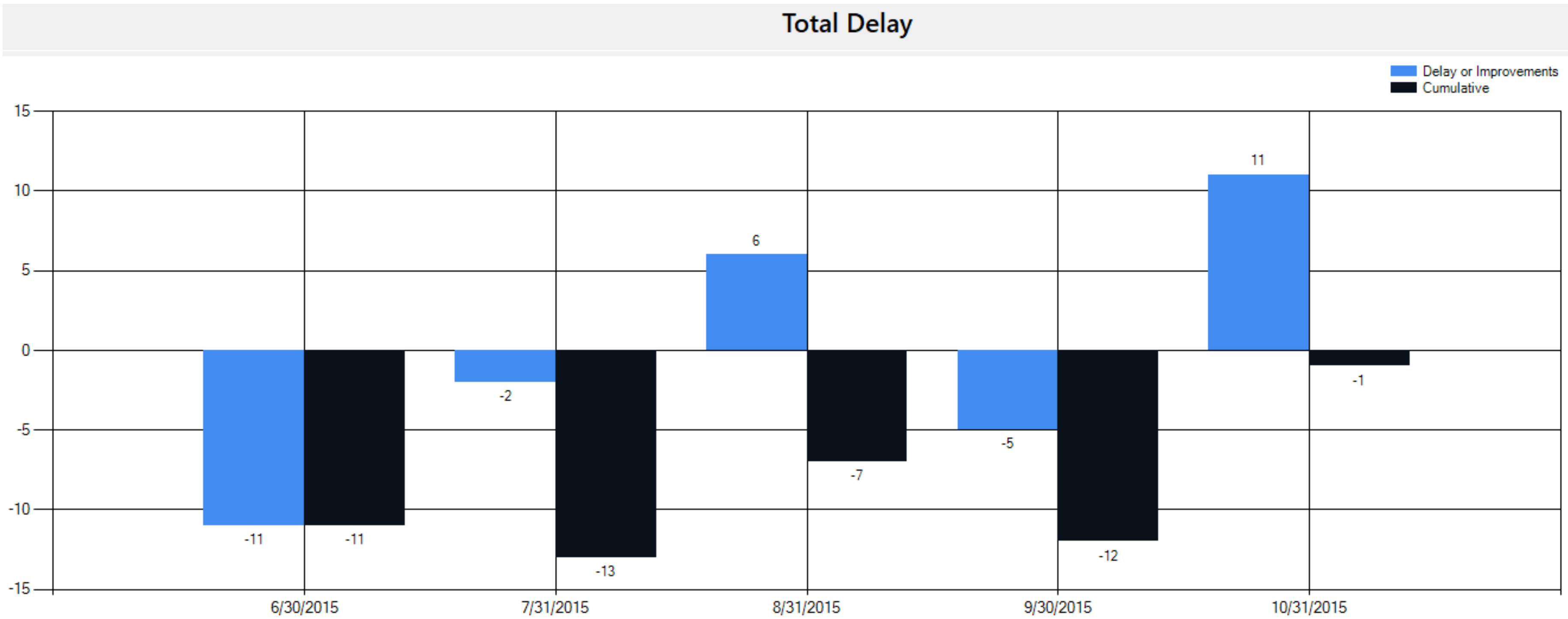


Figure 6.28 Output table of MSvsMUS add-in tool of Bexel Manager

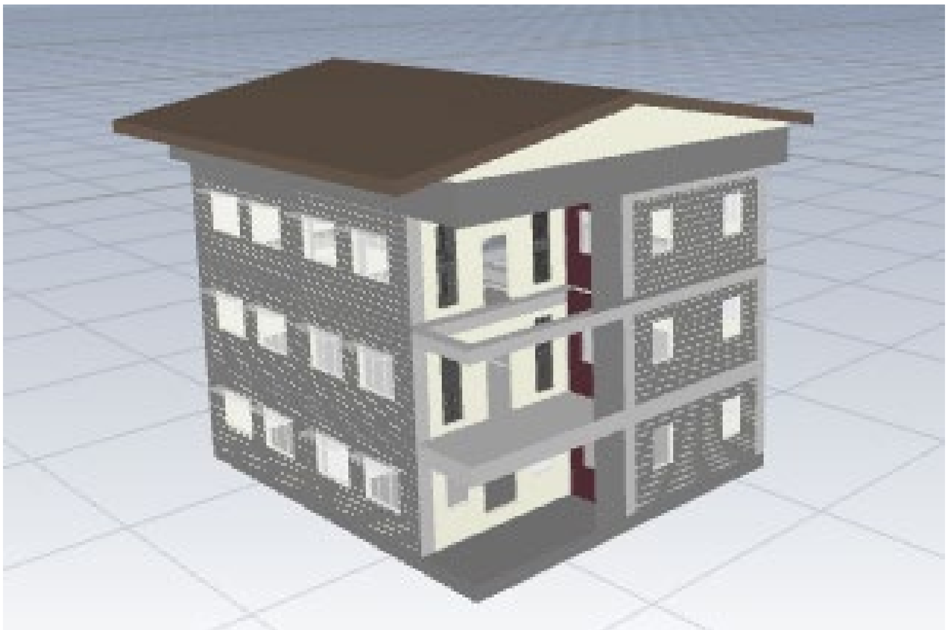


Figure 6.29 Representation of delayed activities on the isometric view of the dwelling project

As is seen in Table Figure 6.25, the add-in tool in Bexel Manager successfully computed and apportioned the delay to the contracting parties by using MSvsMUS method. 11 day-improvement was made by the contractor according to“Early and late Commencement Date”. The owner delayed 12 days according to “Delays of ECDs and ENDs”.

7
DISCUSSIONS

Resolution of delay-related dispute requires a mutual agreement in terms of defining the responsible parties and extension of time. However, identifying responsible parties for delays in construction projects is the most likely source of dispute. Although numerous studies concerning the delay resolution in the construction industry have been published over the past few decades, the construction industry still suffers from construction delays [118]; therefore, this thesis was conducted in order to improve delay resolution process. Discussion section of this study concerning problem justifications are presented under two sections which are "Scientometric Analysis" and "Unstructured Interviews" respectively. Afterwards, the developed method concerning delay analysis is provided within the section of "Development of Delay Analysis Model (MSvsMUS)" in this section. Finally, integration of developed method into Bexel Manager which is 4D-BIM software is demonstrated under the section, namely "Integration of MSvsMUS into BIM Software as Add-In Tool".

7.1 Scientometric Analysis

The problems, gabs and trends concerning the current delay analysis methods in the construction industry were detected. Although many research efforts dealing with different aspects of delay analysis have been made, there has been no study summarizing the studies conducted so far and identifying new study areas related to this subject. Therefore, first section of the problem detection in this thesis is extremely valuable for drawing the road map for the examiners who aim to work in this field. It is believed that the overarching outcome of this study is as much a significant guide for the researchers as it is for the practitioners.

In this section, the keywords concerning delay analysis subjects were detected and the documents published between 1982 and 2020 (15th November) were searched through Scopus Database. After an in-depth elaboration, the irrelevant researches were omitted, and thus 168 documents were detected. Later, scientometric analysis such as ‘Co-Item Analysis’ was conducted to determine the trend and gaps in the literature concerning the delay analysis domain.

In this section, main research domains, gaps and trends concerning delay analyses were released via Co-Item Analysis. Co-Item Analysis revealed that themes such as technique, approach, delay analysis method, model, concurrent delay, delay analysis technique, case, study, cause, effect, impact and factor took the attention of the researchers for resolving the problems resulting from delays in construction projects. When the keywords were elaborated as per publication years, it seems that the research subject concerning improving the delay analysis techniques is a more current issue than resolving the dispute before its occurrence. Due to the downsides of the existing delay analysis methods, their techniques and effectiveness were detected as the most frequently mentioned and widely discussed themes by the researchers [14], [15], [17]–[23], [136]. Despite the numerous contributions made by the researchers, the delay issue remains one of the biggest concerns of the construction industry and most of the projects still suffer from dispute resolution in terms of time. Perera & Sutrisna [38] emphasize that “*The outcomes produced by some methods are more reliable and relatively objective, but those by others are overwhelmingly subjective. However, none of these methods is considered perfect.*” It is clear that the techniques concerning delay analysis require to be improved; therefore, today, improving the delay analysis models is one of the most up to date subjects for the academicians. One of the most striking problems concerning the existing delay analysis models identified by the researchers is ignoring the CPM during the computation of delay and designation of it to the responsible parties [40], [41]. Additionally, actualized schedules may contradict with the planned schedule [43]; therefore, disregarding the actual site progress cannot give the accurate result for delay analyses [44], [45]. Actual progress consideration for the time intervals is also vital to detect the fluctuations on the CP during the execution of delay analysis [28], [40], [46]. Delays are not always successive on the CP, from time to time, delays resulting from owner and contractor can be concurrent; therefore, these delays should be detected and apportioned to the parties [8], [20], [136], [140] in line with the float ownerships defined in the project contact [14], [20], [40], [46], [141], [145], [146]. Additionally, many of the delay analysis models compute the delays for only one side of the party at a time, and this requires extra computation along with associated time and cost. Not only the delayed activities but also time-shortened activities are to be considered during the execution of the delay analysis [2], [40]. Also, the accuracies of delay analysis methods are questioned by practitioners and courts [40]. The most accepted delay analysis methods in the construction industry such as Time Impact Analysis, But-For Analysis, Windows Analysis and their derivatives [16], [17], [40], [44] entail excessive number of requirements such as time, cost and experts [14], [34], [37], [44], [136], [143], [147], [148]; therefore, difficulties of their implementations make them less favorable in the industry [17]. It can be derived from the existing studies that the most appropriate delay analysis models should consider the following attributes; CPM, actual sequence of work, concurrency, float ownership and fluctuations of critical paths. Researches concerning the benchmarking of the existing delay analysis models to examine their strengths and weaknesses over these attributes can be conducted and improvement in the models can be sought. Another flaw of the existing delay analyses - which also triggers the institutions to develop new models - is the inconsistent applications of the delay analysis models in the construction industry due to the lack of the procedures [21], [47]. Institutions like Association for the Advancement of Cost Engineering (AACE) and Society of Construction Law (SCL) contributed to the development of delay resolution process. In 2006, AACE published AACE International Recommended Practice No. 52R-06 analysis [143]. This practice provides a guideline without establishing a standard and discusses the application of Time Impact Analysis in the construction industry. In 2007, AACE published AACE International Recommended Practice No. 29R-03 Forensic Schedule Analysis [149]. The purpose of this practice is to deliver a unifying reference concerning the technical principles and guidelines for CPM applications in the construction industry to conduct a forensic schedule analysis. Additionally, this practice discusses delay analysis methods, irrespective of whether delay analysis methods are acceptable or unacceptable by courts. AACE published 'AACE Recommended Practice for Forensic Schedule Analysis' in 2010 [70]. This practice refers to investigating the events by adopting the CPM or the other recognized scheduling methods in conjunction with the process concerning the delay resolution and other concerning legal issues. Following this, SCL published 'Society of Construction Law Delay and Disruption Protocol' in 2017 in order to offer useful guidance for delay and disruption issues occurring in construction projects [44]. Although there are such recommended practices and protocols in order to maintain consistent applications of delay analyses in the construction industry, it is suggested that algorithms and software integration are to be developed for the most accepted delay analysis models. In literature, most of the delay analysis methods are described via Gantt Charts; however, procedure and algorithms considering the relationships of activities are needed. Thus, the application of delay analysis in construction projects can be standardized and disputes arising from different applications can be avoided.

Although delay analyses are mostly conducted in planning tools, due to the increasing demand for BIM in the construction industry [150], 4D-BIM software tends to replace planning software in the near future. There have been researches that endeavour to improve the process of delay analysis implementations in BIM. The applicability of the delay analysis in BIM has been discussed by the researchers and it has been concluded that most of the problems encountered during the delay analysis can be overcome through integration of BIM [53], [54]. An add-in tool, which is capable of conducting 'Impacted As-Planned' delay analysis model, was developed into BIM software (Revit 2017) by using C programming language [50]. Vacanas et al. [52] proposed accommodating Unmanned Aerial Vehicle (UAV) technologies in infrastructure projects for delay and disruption analyses. In addition to this, some fruitful techniques and frameworks for computation of delays via BIM software have been presented in the literature [55], [118], [151]. Marzouk et al. [55] proposed a methodology which visualizes and foresees claims to assist project parties in handling such claims in a proactive manner by combining 5D BIM model with responsibility matrix of claims' causes. Essawy and Nassar [152] presented a model and software identifying the topological relationship of each building element type so as to generate construction sequence and its associated schedule. Marzouk et al. [59] proposed a model which illustrates how BIM-based claims can be analysed. Integration of delay analysis methods into 4D-BIM is observed to be useful for the areas such as partnerships, negotiations, third party neutrals, referees, litigation process, claims metrics in relation to the use of 4D simulation, value of the claim and 4D simulation of delayed activities; therefore projects may gain efficiency with the use of BIM [153]. Additionally, major challenges occurring during the implementation of delay analyses such as obtaining information and presenting the delay events can be overcome through BIM implementations [51]. Based on the aforementioned studies, offering an information-intensive visualization, BIM is believed to be an ideal means to solve the problems encountered in delay analysis domain. Although BIM provides visualization of delayed events, which is preferred by the stakeholders, undeniable challenges may also occur during the course of the delay analysis process via BIM software. Soltani et al. [154] conducted in-person interviews with construction law attorneys and forensic engineers and spotted these challenges as the high cost of BIM implementation, time consumption, preference of conventional tools for the uncomplicated delays, complexity of BIM for judicial actors such as judges and expert witnesses, unawareness of judicial actors concerning BIM, insufficiency of BIM tools to develop trustworthy 3D models and potential of BIM to change the real story. Despite the expected effectiveness of BIM-based delay analysis, it is derived from the studies that there are some challenges encountered during the course of BIM adaptation in construction industry [155]; therefore, benefits of 4D-BIM based delay analysis tools in practice can be set forth by researchers in order to raise awareness for the construction actors.

This study is unique in several ways. This paper presents the first comprehensive scientometric study investigating researches on delay analysis, which is believed to shed light on the delay analysis context, rather than focusing on a case or an application. In other words, this study is believed to be invaluable in terms of providing a demonstration of previous research efforts to guide future scientific attempts concerning delay analysis domain. In practical terms, this study can enable the practitioners to synthesize delay analysis studies that capture the state-of-the-art researches in delay analysis in the construction industry. Furthermore, this study provides a benchmarking tool for the practitioners to assess the delays occurring in the construction projects and also enhances the readiness of practitioners for adopting delay analysis methods and associated technology. Although many scientific studies have been published in different aspects of the delay analysis domain, projects suffer from the delays occurring during the course of the projects. Concerning the delay analysis methods, TIA, Windows Analysis and their derivatives are the most adopted methods by the practitioners and courts. However, there is no practice illuminating the application techniques of these methods. The delay analysis model mentioned in a construction contract may result in conflict between the contractual parties in terms of the application of the concerning delay analysis. Hence, it is recommended that standards and procedures of delay analysis methods are to be published for the sake of the construction industry. This will improve the construction process and save time and cost by reducing the time-related disputes. In addition to this, many authors detected the defects of the existing delay analysis models; therefore, a delay analysis method overcoming the defects of the existing delay analysis methods can be developed. Furthermore, few researchers emphasized the importance of the delay analysis application in Building Information Modelling (BIM). Delay analysis methods are dependent on scheduling software; however, there is an upsurging trend towards using 4D-BIM software instead of scheduling software in the construction industry. So, it is highly recommended that delay analysis models have to be transferred to 4D-BIM software. In addition to this, by scientometric analysis, this study set forth the pitfalls of the current delay analysis models. This is believed to raise the awareness of the practitioners and help the practitioners to choose the most appropriate delay analysis models for their projects. Detected problems, gaps and trends are also believed to pave the way for new research domains. Notwithstanding the contributions of this scientometric study, it has a limitation. Although VOSviewer, which is a software tool for constructing and visualizing bibliometric networks, is one of the most preferable scientometric tools, additional analyses that are not supported by VOSviewer software can be conducted via other scientometric tools in order to reveal further outcomes concerning delay analysis domain.

7.2 Unstructured Interviews

Unstructured interviews were conducted with 7 experts experienced in the domain of delay analysis. Mainly 3 major drawbacks encountered during the course of the practices of delay analyses in the construction industry were highlighted during the unstructured interview. These are computing any further delay and improvement made by the contractor on the activities of fragnet, an overwhelming number of requirements of the delay analyses, lacking the procedure of the delay analysis. A new delay analysis computing any further delay and improvement made by the contractor on the activities of fragnet, and having a procedure with its algorithm is believed to speed the delay analysis process as well as the reducing the required man-hours.

7.3 Development of Delay Analysis Model (MSvsMUS)

Although delay analysis methods have been created for systematic assessments of delayed activities to settle the project debates concerning time, it is clearly inferred from this study that the detected pitfalls of concerning delay analysis methods are likely to lead to some unenviable outcomes in the construction industry.

By diminishing the inadequacies in the process of the construction sector concerning delay resolution, this study is believed to be unique in several ways. Firstly, this method is able to compute any delays and improvements made by contractors for planned ECDs and ENDs. As is stated in the literature, retrospective delay analysis methods such as Windows Delay Analysis, Collapse As-Built Analysis, As-Planned vs As-Built Analysis and their variants are capable of computing actual delays concerning the subnets created due to ECDs and ENDs

[31]. Prospective delay analysis methods such as Time Impact Analysis, Impacted As-Built Analysis and their variants are capable of computing planned delays concerning the subnets created due to ECDs and ENDs [15]. Despite the capabilities of the existing delay analysis methods, as is explained in the manuscript, MSvsMUS is capable of computing any delays and improvements made by contractors for planned ECDs and ENDs by comparing both planned and actualized ECDs and ENDs, which constitutes the superiority of this method among the current delay analysis methods by presenting an unprecedented outcome. Additionally, conducting delay analysis through Gantt Charts without considering the relationships of activities is likely to result in inaccuracy in the applications of delay analysis methods. In the literature, only FS relationship between the activities are considered and the relationships between the activities such as SS and FF are not considered during the explanations and validations of WA, derivatives of WA, But-For Delay Analysis, derivatives of But-For Delay Analysis and TIA [15], [30], [34], [40], [140], [156], [157]. However, this presents limited practical implications since activities having SS and FF relationships are very common in construction projects. Thus, delay analysis methods which do not consider these relationships are not likely to be fully successful during implementation in real construction projects. In contrast to the other delay analysis methods, this proposed method takes the relationships – which are FS, FF and SS - of the activities into account during the calculation of the delays. Moreover, as is adopted by the most accepted delay analysis methods such as WA, TIA and their derivatives, MSvsMUS also considers the actual sequence of work, CP and fluctuations on CP thanks to the periodical analysis of CP. Furthermore, most of the techniques of delay analysis models are described without any given numerical definitions, which may result in ambiguities in their practical applications. On the other hand, MSvsMUS provides a complete procedure and pseudocodes; therefore, this can prevent erratic implementations in practice and can facilitate the method to be transferred into the delay analysis software. On the whole, it is explicitly seen that MSvsMUS overcomes the major drawbacks defined in the section of “Principles of the developed method” in this

study. These advantages constitute the strengths of this method and it is believed to prevail against the existing delay analysis methods in terms of these features. Based on the conclusion obtained from this study, MSvsMUS has been found out to be a good alternative for delay analyses when the baseline schedule, updated schedules and subnets including ECDs and ENDs are available.

7.4 Integration of MSvsMUS into BIM Software as Add-In Tool

The widespread use of 4D-BIM in the construction industry has been increasing in the construction industry. Since delay analysis can only be conducted with planning software; construction firms accommodate both analyst and software concerning BIM and planning. It is very complicated to conduct the scheduling tasks and delay analysis on different platforms, and it brings many problems and challenges in practice. Additionally, one of the detected problems obtained from the scientometric analysis was the lack of delay analysis implementation in BIM domain. Even though many projects adopt 4D-BIM for scheduling tasks, current 4D-BIM tools are not capable of implementing the preferred delay analysis methods such as Time Impact and Windows Analysis. Therefore, this study integrated MSvsMUS delay analysis method into 4D-BIM software, namely Bexel Manager as an add-in tool. The software was validated successfully via a case study which is a real dwelling project. By comparing the baseline schedule and the last schedule update which is Modified Updated Schedule for Window 5 in the hypothetical study, it can be inferred that project was delayed for one day. As is detected from the outcomes of the case study which is a real dwelling project analyzed by MSvsMUS, there is only one day delay in the project. An 11 day-improvement and a 12 day-delay were made by the contractor and owner respectively in the project. It is concluded that integrated MSvsMUS into Bexel Manager was able to compute the project delay and apportion the project delays to the project parties successfully.

8

CONCLUSION

Construction projects which are unique and sophisticated in nature are prone to changes during the course of the project life cycle. Changes mostly result in conflicts concerning time and cost in projects and most of the time they turn into disputes over time. Dispute – which is one of the inevitable parts of the construction projects – has to be settled otherwise it may end up with further time and cost losses as well as deteriorate relationships of the contracting parties. Being one of the most common dispute causes in construction projects, delay is extremely important to be resolved in a timely manner by using the most suitable delay analysis methods. In construction projects, identifying the responsible parties for delays, which requires a mutual agreement, is the most likely source of dispute. The concept of delay analysis has received rising global attention from institutes, researchers and practitioners due to its significance in the construction industry. In order to define the specific research areas as well as current drawbacks with respect to delay analysis domain in the construction industry, the relevant data were collected via in-depth literature review. Considering the findings of this study with respect to the drawbacks of the existing delay analysis methods, it has been detected that the following major problems exist within the delay analysis domain in the literature.

- None of the existing delay analysis methods compute any further delays or improvements made by the contractor for the fragnet/s.
- All of the existing delay analysis methods ignore Finish to Start, Start to Start and Finish to Finish relationships of the activities in the project schedule during their demonstrations in the literature.
- There is no study concerning delay analysis integration into 4D-BIM software for the most adopted delay analysis methods using Critical Path Method and considering actual site progress.

- Some of the existing delay analysis methods don't analyze critical paths in the project schedule.
- Only concentrating on the planned schedule, some of the existing delay analysis methods ignore actual site progress.
- Some of the existing delay analysis methods ignore fluctuations on the critical path in the project schedule.
- Some of the existing delay analysis methods don't involve a numerical explanation.
- There is no integrated software concerning delay analysis methods for construction projects.
- Some of the delay analysis methods consider only the owner's delays without computing the contractor's delay, which may end up with conflict between the project parties.
- Most of the delay analysis methods don't have any procedure to ensure their consistent practical applications in the construction projects.

It is clear that conducting a delay analysis disregarding the detected shortcomings cannot be considered as reliable. In order to justify the abovementioned drawbacks with respect to the delay analysis methods in the construction industry, the relevant data were collected via in-depth literature review and an expert review meeting conducted with 7 experts experienced in delay analyses. Afterward, in this study, a new delay analysis, namely MSvsMUS was developed and this method was applied in a hypothetical study to compare the outputs of the MSvsMUS with Windows Delay Analysis method which is one of the most adopted delay analysis methods by the practitioners and courts. It was deduced from this study that MSvsMUS has superiority over Windows Delay Analysis method and its success was validated via the hypothetical study. Later, MSvsMUS was presented with a case study which is a real airport project. Next, in order to further validate the developed method, the implementation of the method and the results obtained from the case study were demonstrated to the same 7 experts and the strengths and weaknesses of this method were discussed in an expert panel. It was concluded that MSvsMUS was successfully implemented and its applicability in the construction industry was confirmed by the experts. Also, experts in the expert panel highlighted that MSvsMUS is believed to be unique in several ways. Firstly, this method is able to compute any delays and improvements made by contractors for planned ECDs and ENDs as opposed to other delay analysis methods. In addition to this, this study considers relationships of the activities including FS, SS and FF which are not adopted by the existing delay analysis methods in the literature except for FS relationships. Moreover, this study considers the actual sequence of work, fluctuations on CP by means of the periodical analysis of CP as opposed to the majority of delay analysis methods.

Surging productivity and efficiency has become the main goal of AEC sector, and the management of information is one of the best ways to accomplish this objective [158]. By the same token, The Architectural, Engineering and Construction (AEC) industry has rapidly changed and adapted itself in line with the developments in technology like Building Information Modeling (BIM) [159]. BIM is one of the most powerful tools concerning project management by enhancing knowledge sharing during the life cycle of structure from the initial design to facility management [160]. BIM technology has become a cornerstone of project management in the construction industry. Therefore, BIM is a comprehensive tool that has been adopted by many projects in AEC industry in the sense that it is based on a virtual 3D model to manage much of the information concerning building life cycle. Moreover, BIM tools offer novelties concerning 4D (Schedule Management). However, when 4D-BIM and scheduling software are compared, it is clearly identified that scheduling software is far ahead of the 4D-BIM software. Detecting the gaps in the 4D-BIM software, numerous researchers have contributed to the development of the body of knowledge concerning BIM to improve the 4D-BIM domain. Although BIM helps to expedite the process of construction projects via its attributes such as rendering, animation, scheduling and budgeting, BIM also brings challenges for the building process as well as involve some drawbacks [161], [162]. Notwithstanding the existing fruitful features improving the project management process, the construction industry still encounters stark obstacles during its implementation. Although great efforts have been made in BIM development, practitioners face various difficulties during the course of the applications of BIM [159], [163]–[165]. In order to diminish one of the major problems of 4D-BIM software, which is the lack of delay analysis feature, this study integrated MSvsMUS delay analysis method into 4D-BIM software which is Bexel Manager as an add-in tool. The developed add-in tool was validated successfully via a case study which is a real dwelling project. The proposed BIM-based delay analysis method in this thesis will be unique and enable the practitioners to use solely BIM software without depending on scheduling software. This attribute of MSvsMUS is believed to be the key strength of this delay analysis method compared with the existing delay analysis methods in the construction industry. This study is also believed to minimize delay-related losses such as time and money in the construction industry and empower the construction sector by enabling project parties to proceed without dispute and clashing caused by delays.

Providing the complete procedure and pseudocodes, MSvsMUS will lay the groundwork for the researchers to conduct future studies which aim to develop software related to the delay analysis domain in order to ensure consistency in construction projects. In other words, the developed method is believed to be precious in that it paves the way for the studies aiming to automate the delay analyses to ensure consistency in construction projects. This software may also be integrated into other applications such as Enterprise Resource Planning (ERP), scheduling and other BIM software by using provided pseudocodes and this is believed to provide a competitive advantage for the welfare of the construction companies. It is highly believed that this will reduce time, cost and manpower of experts, and maintain consistent practical applications as well as provide a sustainable advantage for the construction companies among their competitors by enhancing the delay analysis process. It is also believed that this method highly contributes to the literature concerning delay and improvement (acceleration and mitigation) analyses in this sense and it is also believed that this study will shed light on future studies concerning delay analyses and lead the projects to be held on a stronger basis in terms of time. Furthermore, pseudo-codes and the codes developed for MSvsMUS which are provided in this thesis can be used for future software that can be either a 4D-BIM software or stand alone delay analysis software. It is belived that this will reduce the dependency of scheduling software in the construction industry, and 4D-BIM is able to resolve all the issues concerning the design and scheduling in the same platform without the need of planning software.

Concerning the limitation of this study, delay analysis such as Windows Delay Analysis method and its variants compute the delays by comparing two successive schedules as per the critical activities detected in the former schedule. Critical path in the former update is likely to change in the latter update due to fluctuations on the critical path. MSvsMUS has the same drawbacks; therefore, it is highly recommended that windows period should be taken as short as possible. This method may be conducted on a daily basis in order to abstain from any fluctuations on the critical path of a schedule; however, it should be noted that updating the schedule on a daily basis requires plenty of resources, close cooperation and strong coordination between the internal and external stakeholders of a project. Hence, developing software by using the pseudocode provided in this study can overcome these bottlenecks. Additionally, concurrency and prospective analysis can be introduced to MSvsMUS. This method is limited to CPM technique. With respect to the schedules based on Line of Balance (LOB), other eligible methods require to be utilized.

REFERENCES

[1] A. Mills, "A Systematic Approach to Risk Management for Construction," *Struct. Surv.*, vol. 19, no. 5, pp. 245–252, 2001, doi: https://doi.org/10.1108/02630800110412615.

[2] Y. Kim, K. Kim, and D. Shin, "Delay Analysis Method Using Delay Section," *J. Constr. Eng. Manag.*, vol. 131, no. 11, pp. 1155–1164, 2005, doi: 10.1061/(ASCE)0733-9364(2005)131:11(1155).

[3] G. Bilgin, I. Dikmen, and M. T. Birgonul, "An Ontology-based Approach for Delay Analysis in Construction," *KSCE J. Civ. Eng.*, vol. 22, no. 2, pp. 384–398, 2018, doi: 10.1007/s12205-017-0651-5.

[4] M. Özdemir, "A Probabilistic Schedule Delay Analysis in Construction Projects by Using Fuzzy Logic Incorporated with Relative Importance Index (RII) Method," Civil Engineering, 2010.

[5] W. N. Salim and C. M. Sujana, "Project Delay Analysis of Highrise Building Project in Jakarta," in *IOP Conference Series: Earth and Environmental Science*, 2020, vol. 426, no. 1, doi: 10.1088/1755-1315/426/1/012051.

[6] G. R. Stumpf, "Schedule Delay Analysis," *Cost Eng.*, vol. 42, no. 7, pp. 32–43, 2000.

[7] B. Zarei, H. Sharifi, and Y. Chaghouee, "Delay Causes Analysis in Complex Construction Projects: A Semantic Network Analysis Approach," *Prod. Plan. Control*, vol. 29, no. 1, pp. 29–40, 2018, doi: 10.1080/09537287.2017.1376257.

[8] S. T. Ng, M. Skitmore, M. Z. M. Deng, and A. Nadeem, "Improving Existing Delay Analysis Techniques for the Establishment of Delay Liabilities," *Constr. Innov.*, vol. 7, no. 1, pp. 3–17, 2004.

[9] S. Alkass, M. Mazerolle, and F. Harris, "Construction Delay Analysis Techniques," *Constr. Manag. Econ.*, vol. 14, no. 5, pp. 375–394, 1996, doi: 10.1080/014461996373250.

[10] E. Viles, N. C. Rudeli, and A. Santilli, "Causes of Delay in Construction Projects: a Quantitative Analysis," *Eng. Constr. Archit. Manag.*, vol. 27, no. 4, pp. 917–935, 2019, doi: 10.1108/ECAM-01-2019-0024.

[11] S. Bektas, M. Talat Birgonul, and I. Dikmen, "Integrated Probabilistic Delay Analysis Method to Estimate Expected Outcome of Construction Delay Disputes," *J. Leg. Aff. Disput. Resolut. Eng. Constr.*, vol. 13, no. 1, pp. 04520037-1–18, 2021, doi: 10.1061/(asce)la.1943-4170.0000439.

[12] M. Sambasivan, T. J. Deepak, A. N. Salim, and V. Ponniah, "Analysis of Delays in Tanzanian Construction Industry Transaction Cost Economics (TCE) and Structural Equation Modeling (SEM) Approach," *Eng. Constr. Archit. Manag.*, vol. 24, no. 2, pp. 308–325, 2017, doi: 10.1108/ECAM-09-

2015-0145.

[13] M. Gündüz, Y. Nielsen, and M. Özdemir, "Quantification of Delay Factors Using the Relative Importance Index Method for Construction Projects in Turkey," *J. Manag. Eng.*, vol. 29, no. 2, pp. 133–139, 2013, doi: 10.1061/(asce)me.1943-5479.0000129.

[14] D. Arditi and T. Pattanakitchamroon, "Selecting a Delay Analysis Method in Resolving Construction Claims," *Int. J. Proj. Manag.*, vol. 24, no. 2, pp. 145–155, 2006, doi: 10.1016/j.ijproman.2005.08.005.

[15] M. Asim, S. Deep, and S. A. Ahmad, "Time Impact Study of Real Estate Sector Construction Projects Post Application of Lean Principles for Delay Resolutions," *Int. J. Civ. Eng. Technol.*, vol. 8, no. 2, pp. 89–99, 2017.

[16] N. Braimah, "Approaches to Delay Claims Assessment Employed in the UK Construction Industry," *Buildings*, vol. 3, no. 3, pp. 598–620, 2013, doi: 10.3390/buildings3030598.

[17] N. Braimah and I. Ndekugri, "Consultants' Perceptions on Construction Delay Analysis Methodologies," *J. Constr. Eng. Manag.*, vol. 135, no. 12, pp. 1279–1288, 2009, doi: 10.1061/(ASCE)CO.1943-7862.0000096.

[18] T. Farrow, "Developments in the Analysis of Extensions of Time," *J. Prof. Issues Eng. Educ. Pract.*, vol. 133, no. July, pp. 218–228, 2007, doi: https://doi.org/10.1061/(ASCE)1052-3928(2007)133:3(218).

[19] K. D. Gothand, "Schedule Delay Analysis: Modified Windows Approach," *Cost Eng.*, vol. 45, no. 9, pp. 18–23, 2003.

[20] S. B. Mohan and K. S. Al-Gahtani, "Current Delay Analysis Techniques and Improvements," *Cost Eng.*, vol. 48, no. 9, pp. 12–21, 2006.

[21] I. Ndekugri, N. Braimah, and R. Gameson, "Delay Analysis within Construction Contracting Organizations," *J. Constr. Eng. Manag.*, vol. 134, no. 9, pp. 692–700, 2008, doi: 10.1061/(ASCE)0733-9364(2008)134:9(692).

[22] J. Bin Yang, K. M. Huang, and C. H. Lee, "Evaluation of Delay Analysis Methodologies on Lost Productivity in Construction Projects," in *2010 - 27th International Symposium on Automation and Robotics in Construction, ISARC 2010*, 2010, pp. 677–683, doi: 10.22260/isarc2010/0072.

[23] J. G. Zack, "But-for Schedules - Analysis and Defense," *Cost Eng. (Morgantown, West Virginia)*, vol. 43, no. 8, pp. 13–17, 2001.

[24] L. D. Nguyen and W. Ibbs, "FLORA: New Forensic Schedule Analysis Technique," *J. Constr. Eng. Manag.*, vol. 134, no. 7, pp. 483–491, 2008, doi: 10.1061/(asce)0733-9364(2008)134:7(483).

[25] O. H. Türkakın, E. Manisalı, and D. Arditi, "Delay Analysis in Construction Projects with no Updated Work Schedules," *Eng. Constr. Archit. Manag.*, vol. 27, no. 10, pp. 2893–2909, 2020, doi: 10.1108/ECAM-09-2019-0470.

[26] A. Mbabazi, T. Hegazy, and F. Saccomanno, "Modified But-For Method for

Delay Analysis," *J. Constr. Eng. Manag.*, vol. 131, no. 10, pp. 1142–1144, 2005, doi: 10.1061/(ASCE)0733-9364(2005)131:10(1142).

[27] J.-S. Lee, "Delay Analysis Using Linear Schedule in Construction," *AACE Int. Trans.*, pp. 1–6, 2007.

[28] T. Hegazy and W. Menesi, "Delay Analysis Under Multiple Baseline Updates," *J. Constr. Eng. Manag.*, vol. 134, no. 8, pp. 575–582, 2008, doi: 10.1061/(ASCE)0733-9364(2008)134:8(575).

[29] J. Bin Yang and Y. L. Teng, "Theoretical Development of Stochastic Delay Analysis and Forecast Method," *J. Chinese Inst. Eng.*, vol. 40, no. 5, pp. 391–400, 2017, doi: 10.1080/02533839.2017.1336115.

[30] J.-B. Yang and P.-C. Yin, "Isolated Collapsed But-For Delay Analysis Methodology," *J. Constr. Eng. Manag.*, vol. 135, no. 7, pp. 570–578, 2009, doi: 10.1061/(ASCE)CO.1943-7862.0000016.

[31] T. Hegazy, E. Elbeltagi, and K. Zhang, "Keeping Better Site Records Using Intelligent Bar Charts," *J. Constr. Eng. Manag.*, vol. 131, no. 5, pp. 513–521, 2005, doi: 10.1061/(ASCE)0733-9364(2005)131:5(513).

[32] J. Bin Yang, C. K. Kao, and Y. Y. Lee, "System Requirement Analysis of a Construction Delay Analysis System," in *2006 Proceedings of the 23rd International Symposium on Robotics and Automation in Construction, ISARC 2006*, 2006, pp. 102–106, doi: 10.22260/isarc2006/0022.

[33] J. Bin Yang and C. K. Kao, "Critical Path Effect Based Delay Analysis Method for Construction Projects," *Int. J. Proj. Manag.*, vol. 30, no. 3, pp. 385–397, 2012, doi: 10.1016/j.ijproman.2011.06.003.

[34] T. Hegazy and K. Zhang, "Daily Windows Delay Analysis," *J. Constr. Eng. Manag.*, vol. 131, no. 5, pp. 505–512, 2005.

[35] I. Adhikaril, S. Kim, and Y. Lee, "Selection of Appropriate Schedule Delay Analysis Method : Analytical Hierarchy Process (AHP)," in *PICMET 2006 Proceedings,* 2006, pp. 483–488, doi: 10.1109/PICMET.2006.296646.

[36] N. Braimah and I. Ndekugri, "Factors Influencing the Selection of Delay Analysis Methodologies," *Int. J. Proj. Manag.*, vol. 26, no. 8, pp. 789–799, 2007, doi: 10.1016/j.ijproman.2007.09.001.

[37] N. A. Perera, M. Sutrisna, and T. W. Yiu, "Decision-Making Model for Selecting the Optimum Method of Delay Analysis in Construction Projects," *J. Manag. Eng.*, vol. 32, no. 5, pp. 1–14, 2016, doi: 10.1061/(ASCE)ME.1943-5479.0000441.

[38] N. Perera and M. Sutrisna, "The Use of Analytic Hierarchy Process (AHP) in the Analysis of Delay Claims in Construction Projects in the UAE," *Built Hum. Environ. Rev.*, vol. 3, no. 1, pp. 29–47, 2010.

[39] I. S. Abotaleb and I. H. El-adaway, "Managing Construction Projects through Dynamic Modeling: Reviewing the Existing Body of Knowledge and Deriving Future Research Directions," *J. Manag. Eng.*, vol. 34, no. 6, p.

04018033, 2018, doi: 10.1061/(asce)me.1943-5479.0000633.

[40] C. K. Kao and J. Bin Yang, "Comparison of Windows-based Delay Analysis Methods," *Int. J. Proj. Manag.*, vol. 27, no. 4, pp. 408–418, 2009, doi: 10.1016/j.ijproman.2008.05.016.

[41] V. Rooholelm and A. Sheikh Aboumasoudi, "Share Determination of Stakeholder Delays, Based on Targeted Delay Analysis of Projects, with incursive and defensive (In-De) approach," *Int. J. Manag. Proj. Bus.*, 2020, doi: 10.1108/IJMPB-08-2019-0196.

[42] R. A. Harris and S. Scott, "UK Practice in Dealing with Claims for Delay," *Eng. Constr. Archit. Manag.*, vol. 8, no. 5–6, pp. 317–324, 2001, doi: 10.1108/eb021192.

[43] G. K. Koulinas, A. S. Xanthopoulos, T. T. Tsilipiras, and D. E. Koulouriotis, "Schedule Delay Risk Analysis in Construction Projects with a Simulation-Based Expert System," *Buildings*, vol. 10, no. 8, p. 134, 2020, doi: 10.3390/buildings10080134.

[44] Society of Construction Law, "The Society of Construction Law Delay and Disruption Protocol," Leicestershire (United Kingdom), 2017. [Online]. Available: www.scl.org.uk.

[45] M. N. Maidi, "Lean Construction and Delay Analysis Input in Delay Mitigation Master Thesis," Hochschule für Technik und Wirtschaft Berlin, 2020.

[46] B. Şinasi, "Shortcomings of Delay Analysis Practices : a Case Study Using an Integrated Approach," Middle East Technical University, 2012.

[47] A. Vasilyeva-lyulina, M. Onishi, and K. Kobayashi, "Delay Analysis Methods for Construction Projects : Mathematical Modelling," *Int. J. Transp.*, vol. 3, no. 1, pp. 27–36, 2015, doi: 10.14257/ijt.2015.3.1.03.

[48] M. Çevikbaş and Z. Işık, "An Overarching Review on Delay Analyses in Construction Projects," *Buildings*, vol. 11, no. 3, pp. 1–25, Mar. 2021, doi: 10.3390/buildings11030109.

[49] Cabinet Office, "Government Construction Strategy," 2011.

[50] Z. Tarek, H. Amir, A. Hoda, H. Ossama, N. Khaled, and Z. Khaled, "BIM-Based Delay Analysis : A Framework," in *Building Tomorrow's Society*, 2018, pp. 1–10.

[51] D.-J. Gibbs, S. Emmitt, K. Ruikar, and W. Lord, "An Investigation into Whether Building Information Modelling (BIM) can Assist with Construction Delay Claims," *International Journal of 3-D Information Modeling*, vol. 2, no. 1. pp. 45–52, 2013, doi: 10.4018/ij3dim.2013010105.

[52] Y. Vacanas, K. Themistocleous, A. Agapiou, and D. Hadjimitsis, "Building Information Modelling (BIM) and Unmanned Aerial Vehicle (UAV) Technologies in Infrastructure Construction Project Management and Delay and Disruption Analysis," in *Proceedings of SPIE - The International Society*

for Optical Engineering, 2015, vol. 9535, doi: 10.1117/12.2192723.

[53] H. Y. Chou and J. Bin Yang, “Preliminary Evaluation of BIM-based Approaches for Schedule Delay Analysis,” in *IOP Conference Series: Materials Science and Engineering*, 2017, vol. 245, no. 6, doi: 10.1088/1757-899X/245/6/062048.

[54] K. M. Anderson, “Leveraging the Power of 4D Models for Analysing and Presenting CPM Schedule Delay Analyses,” *AACE Int. Trans.*, vol. 51, no. 3, pp. 49–51, 1986.

[55] M. Marzouk, A. Othman, M. Enaba, and M. Zaher, “Using BIM to Identify Claims Early in the Construction Industry: Case Study,” *J. Leg. Aff. Disput. Resolut. Eng. Constr.*, vol. 10, no. 3, 2018, doi: 10.1061/(ASCE)LA.1943-4170.0000254.

[56] H. Y. Chong, ; Rosli, M. Zin, and S. C. Chong, “Employing Data Warehousing for Contract Administration: e-Dispute Resolution Prototype,” *J. Constr. Eng. Manag.*, vol. 139, no. 6, pp. 611–619, 2013, doi: 10.1061/(ASCE)CO.1943-7862.0000642.

[57] S. Alshammari, K. Al-Gahtani, I. Alhammad, and N. Braimah, “A Systematic Method to Analyze Force Majeure in Construction Claims,” *Buildings*, vol. 7, no. 4, pp. 1–22, 2017, doi: 10.3390/buildings7040115.

[58] P. Shahsavand, A. Marefat, and M. Parchamijalal, “Causes of Delays in Construction Industry and Comparative Delay Analysis Techniques with SCL Protocol,” *Eng. Constr. Archit. Manag.*, vol. 25, no. 4, pp. 497–533, 2018, doi: 10.1108/ECAM-10-2016-0220.

[59] M. M. Marzouk, A. A. Othman, M. M. Enaba, and M. M. Zaher, “Proactive Construction Claims Analysis Using BIM,” *Leadersh. Sustain. Infrastruct.*, pp. 1–10, 2017.

[60] G. Ellis, “100+ Construction Industry Statistics,” *Autodesk Construction Cloud*, 2021. https://constructionblog.autodesk.com/construction-industry-statistics/ (accessed Jan. 14, 2022).

[61] W. Ibbs, “Thinking about Delay, Disruption, and the Cumulative Impact of Multiple Changes,” *J. Leg. Aff. Disput. Resolut. Eng. Constr.*, vol. 5, no. 3, pp. 109–112, 2013, doi: 10.1061/(asce)la.1943-4170.0000126.

[62] N. Braimah, “Understanding Construction Delay Analysis and the Role of Preconstruction Programming,” *J. Manag. Eng.*, vol. 30, no. 5, 2014, doi: 10.1061/(ASCE)ME.1943-5479.0000216.

[63] A. S. Harle, “Delay Analysis in Residential Project by Using Genetic Algorithm,” *Int. J. Adv. Sci. Res. Eng. Trends*, vol. 5, no. 2, pp. 43–48, 2020.

[64] Y. El Hakim, “Analysis of Concurrent Delays in the Construction Industry,” The American University in Cairo, 2020.

[65] S. Nalawade, J. R. Divate, M. V Gaikwad, M. M. Shinde, and R. R. Issac, “Delay Management In Construction Sector Using Microsoft Project,” *Int.*

Res. J. Mod. Eng. Technol. Sci., vol. 2, no. 5, pp. 179–181, 2020.

[66] S. H. Khahro and Z. A. Memon, "Non Excusable Delays in Construction Industry: A Causal Study," *Eng. Technol. Appl. Sci. Res.*, vol. 8, no. 6, pp. 3561–3564, 2018, doi: 10.48084/etasr.2335.

[67] G. Agyekum-Mensah and A. D. Knight, "The Professionals' Perspective on the Causes of Project Delay in the Construction Industry," *Eng. Constr. Archit. Manag.*, vol. 24, no. 5, pp. 828–841, 2017, doi: 10.1108/ECAM-03-2016-0085.

[68] I. H. El-adaway, R. A. Vance, and I. S. Abotaleb, "Understanding Extension of Time under Different Standard Design-Build Forms of Contract," *J. Leg. Aff. Disput. Resolut. Eng. Constr.*, vol. 12, no. 1, p. 04519031, 2020, doi: 10.1061/(ASCE)LA.1943-4170.0000331.

[69] J. Zack, "Pacing Delays-the Practical Effect," *Cost Eng.*, vol. 42, no. 7, pp. 23–28, 2000.

[70] M. F. D. Onofrio and K. P. Hoshino, "AACE Recommended Practice for Forensic Schedule Analysis," Hilton Austin, Austin, Texas, 2010.

[71] T. Hegazy, "EasyPlan Project Management System," 2019. .

[72] M. S. Hashem M. Mehany and N. Grigg, "Delay Claims in Road Construction: Best Practices for a Standard Delay Claims Management System," *J. Leg. Aff. Disput. Resolut. Eng. Constr.*, vol. 8, no. 3, p. 02516001, 2016, doi: 10.1061/(asce)la.1943-4170.0000186.

[73] A. A. Bubshait and M. J. Cunningham, "Comparison of Delay Analysis Methodologies," *J. Constr. Eng. Manag.*, vol. 124, no. 4, pp. 315–322, 1998, doi: https://doi.org/10.1061/(ASCE)0733-9364(1998)124:4(315).

[74] E. M. K. Ekanayake and B. A. K. S. Perera, "Appropriate Delay Analysis Techniques to Analyse Delays in Road Construction Projects in Sri Lanka," *Built Environ. Proj. Asset Manag.*, vol. 6, no. 5, pp. 521–534, 2016, doi: 10.1108/BEPAM-08-2015-0039.

[75] A. Burr, *Delay and Disruption in Construction Contracts*, 5th ed. New York: Informa Law from Routledge, 2016.

[76] J. J. Shi, S. O. Cheung, and D. Arditi, "Construction Delay Computation Method," *J. Constr. Eng. Manag.*, vol. 127, no. 1, pp. 60–65, 2001, doi: https://doi.org/10.1061/(ASCE)0733-9364(2001)127:1(60).

[77] R. Sherko, "A BIM Based Design Optimization Framework for the Energy Efficient Building Design in Turkey," Hasan Kalyoncu University, 2018.

[78] R. Charef, H. Alaka, and S. Emmitt, "Beyond the Third Dimension of BIM: A Systematic Review of Literature and Assessment of Professional Views," *J. Build. Eng.*, vol. 19, no. May 2020, pp. 242–257, 2018, doi: 10.1016/j.jobe.2018.04.028.

[79] C. Eastman, P. Teicholz, R. Sacks, and K. Liston, *A Building Information Management (BIM) Framework and Supporting Case Study for Existing*

Building Operations, Maintenance and Sustainability, 2nd ed. John Wiley & Sons, 2011.

[80] AEC (UK), *AEC (UK) BIM Technology Protocol Practical implementation of BIM for the UK Architectural, Engineering and Construction (AEC) industry.* 2015.

[81] J. Crowther and S. O. Ajayi, "Impacts of 4D BIM on Construction Project Performance," *Int. J. Constr. Manag.*, pp. 1–14, 2019, doi: 10.1080/15623599.2019.1580832.

[82] O. Özkoç, "Changing Role(s) of The Profession of Architecture: Building Information Modeling in Practice," Middle East Technical University, 2015.

[83] M. Sakin, "Development of BIM Implementation Framework for Digital Construction in Turkey," Hasan Kalyoncu University, 2019.

[84] H. I. Aidibi, "Studying the Effect of BIM on Construction Conflicts and Disputes Using Agent-based Modeling," American University of Beirut, 2016.

[85] A. A. A. Azis, A. H. Memon, I. A. Rahman, Q. B. A. I. Latif, and S. Nagapan, "Cost Management of Large Construction Projects in South Malaysia," in *ISBEIA 2012 - IEEE Symposium on Business, Engineering and Industrial Applications*, 2012, no. September, pp. 625–629, doi: 10.1109/ISBEIA.2012.6422964.

[86] M. M. Tahir, N. A. Haron, A. H. Alias, A. N. Harun, I. B. Muhammad, and D. L. Baba, "Improving Cost and Time Control in Construction Using Building Information Model (BIM): A Review," *Pertanika J. Sci. Technol.*, vol. 26, no. 1, pp. 21–36, 2018.

[87] M. Dobelis, "Drawbacks of BIM Concept Adoption," *Sci. Proc. 12th Int. Conf. Eng. Graph. BALTGRAF 2013*, pp. 57–67, 2013.

[88] C. Chen and L. Tang, "BIM-based Integrated Management Workflow Design for Schedule and Cost Planning of Building Fabric Maintenance," *Autom. Constr.*, vol. 107, no. August, p. 102944, 2019, doi: 10.1016/j.autcon.2019.102944.

[89] A. A. Wildenauer, "Critical Assessment of the Existing Definitions of BIM Dimensions on the Example of Switzerland," *Int. J. Civ. Eng. Technol.*, vol. 11, no. 4, pp. 134–151, 2020.

[90] E. O. Alenazi, "Using BIM for Delay Management in Public Sector Construction Projects in KSA," Loughborough University, 2018.

[91] S. M. Khoshnava, A. Ahankoob, R. Rostami, and C. Preece, "Potential Application of BIM in Construction Dispute and Conflict," in *Management in Construction Research Association (MiCRA)*, 2014, pp. 2–7.

[92] D.-J. Gibbs, "Development of Building Information Models (BIM) to Support Innovative Time Management and Delay Analysis," Loughborough University, 2016.

[93] B. YER, “BIM Uygulamalarinda Sözleşmelerden Kaynaklanan Sorunlarin Incelenmesi,” İstanbul Teknik Üniversitesi, 2017.

[94] A. Koutamanis, “Dimensionality in BIM: Why BIM cannot have more than Four Dimensions?,” *Autom. Constr.*, vol. 114, no. February, pp. 1–5, 2020, doi: 10.1016/j.autcon.2020.103153.

[95] C. W. Ibbs, D. B. Ashley, and A. M. Asce, “Impact of Various Construction Contract Clauses,” *J. Constr. Eng. Manag.*, vol. 113, no. 3, pp. 501–521, 1987.

[96] A. Valavanoglou and D. Heck, “Building Information Modeling and Forensic Analysis of Delay and Disruption,” *Proc. Int. Struct. Eng. Constr.*, vol. 3, no. 1, pp. 527–532, 2020, doi: 10.14455/isec.res.2016.50.

[97] M. Çevikbaş and Z. Işık, “Determination and Ranking of Obstacles in BIM Implementation with Relative Importance Index,” in *6th international Project and Construction Management Conference (IPCMC2020)*, 2020, no. November, pp. 12–14.

[98] B. Ch, A. Goud K, A. Prakash, and K. Mounika, “Schedule Delay Analysis of a Multi-storied Residential Project Using Primavera,” in *IOP Conference Series: Materials Science and Engineering*, 2021, vol. 1091, pp. 1–8, doi: 10.1088/1757-899X/1091/1/012038.

[99] Y. Zhang *et al.*, “How Does Experience with Delay Shape Managers’ Making-Do Decision: Random Forest Approach,” *J. Manag. Eng.*, vol. 36, no. 4, p. 04020030, 2020, doi: 10.1061/(asce)me.1943-5479.0000776.

[100] B. Mulholland and J. Christian, “Risk Assessment in Construction Schedules,” *J. Constr. Eng. Manag.*, vol. 125, no. 1, pp. 8–15, 1999, doi: 10.1061/(ASCE)0733-9364(1999)125:1(8).

[101] C. Budayan, I. Dikmen, and M. Talat Birgonul, “Alignment of Project Management with Business Strategy in construction: evidence from the Turkish contractors,” *J. Civ. Eng. Manag.*, vol. 21, no. 1, pp. 94–106, 2015, doi: 10.3846/13923730.2013.802737.

[102] A. A. Abu Hammad, S. M. A. Ali, G. J. Sweis, and R. J. Sweis, “Statistical Analysis on the Cost and Duration of Public Building Projects,” *J. Manag. Eng.*, vol. 26, no. 2, pp. 105–112, 2010, doi: 10.1061/(asce)0742-597x(2010)26:2(105).

[103] C. Chen, “Science Mapping: A Systematic Review of the Literature,” *J. Data Inf. Sci.*, vol. 2, no. 2, pp. 1–40, 2017, doi: 10.1515/jdis-2017-0006.

[104] G. Derbe, Y. Li, D. Wu, and Q. Zhao, “Scientometric Review of Construction Project Schedule Studies: Trends, Gaps and Potential Research Areas,” *J. Civ. Eng. Manag.*, vol. 26, no. 4, pp. 343–363, 2020.

[105] A. A. Chadegani *et al.*, “A Comparison between Two Main Academic Literature Collections: Web of Science and Scopus Databases,” *Asian Soc. Sci.*, vol. 9, no. 5, pp. 18–26, 2013, doi: 10.5539/ass.v9n5p18.

[106] M. Thelwall, “Dimensions: A Competitor to Scopus and the Web of Science?,” *J. Informetr.*, vol. 12, no. 2, pp. 430–435, 2018, doi: 10.1016/j.joi.2018.03.006.

[107] A. V. O. Oliveros and A. R. Fayek, “Fuzzy Logic Approach for Activity Delay Analysis and Schedule Updating,” *J. Constr. Eng. Manag.*, vol. 131, no. 1, pp. 42–51, Jan. 2005, doi: 10.1061/(ASCE)0733-9364(2005)131:1(42).

[108] S. H. Han, S. Yun, H. Kim, Y. H. Kwak, H. K. Park, and S. H. Lee, “Analyzing Schedule Delay of Mega Project: Lessons Learned from Korea Train Express,” *IEEE Trans. Eng. Manag.*, vol. 56, no. 2, pp. 243–256, 2009, doi: 10.1109/TEM.2009.2016042.

[109] D. W. Bordoli and A. N. Baldwin, “A Methodology for Assessing Construction Project Delays,” *Constr. Manag. Econ.*, vol. 16, no. 3, pp. 327–337, 1998, doi: 10.1080/014461998372358.

[110] S. Alkass, M. Mazerolle, E. Tribaldos, and F. Harris, “Computer Aided Construction Delay Analysis and Claims Preparation,” *Constr. Manag. Econ.*, vol. 13, pp. 335–352, 1995, doi: 10.1080/01446199500000038.

[111] A. Darko, A. P. C. Chan, M. A. Adabre, D. J. Edwards, M. R. Hosseini, and E. E. Ameyaw, “Artificial Intelligence in the AEC industry: Scientometric Analysis and Visualization of Research Activities,” *Autom. Constr.*, vol. 112, no. May 2019, p. 103081, 2020, doi: 10.1016/j.autcon.2020.103081.

[112] Gephi, “Gephi - The Open Graph Viz Platform.” .

[113] CiteSpace, “CiteSpace: Visualizing Patterns and Trends in Scientific Literature.” .

[114] R. S. Bankar and S. R. Lihitkar, “Science Mapping and Visualization Tools Used for Bibliometric and Scientometric Studies: A Comparative Study Science Mapping and Visualization Tools Used for Bibliometric and Scientometric Studies View project,” *J. Adv. Libr. Sci.*, vol. 6, no. 1, pp. 382s-394s, 2019.

[115] M. Wang, C. C. Wang, S. Sepasgozar, and S. Zlatanova, “A Systematic Review of Digital Technology Adoption,” *Buildings*, vol. 2020, no. 10, pp. 1–29, 2020, doi: doi:10.3390/buildings10110204.

[116] R. Jin, J. Zuo, and J. Hong, *Scientometric Review of Articles Published in ASCE’s Journal of Construction Engineering and Management from 2000 to 2018*. Turkey, 2017.

[117] BEXEL Manager, “Application Programming Interface (API) - Bexel Manager,” *Bexel Manager*, 2021. https://bexelmanager.com/bexel-manager-api/ (accessed May 19, 2021).

[118] A. S. Askari, D. Porgues, and A. Francis, “Improving Construction Claim Management Using Building Information Model (BIM),” in *Proceedings, Annual Conference - Canadian Society for Civil Engineering*, 2013, pp. 224-1–10.

[119] N. J. van Eck and L. Waltman, "VOSviewer Manual," *Univeristeit Leiden*, 2020. .

[120] S. Aishwarya, D. Muthu, and S. C. Venkata, "Progress Monitoring in a Real Time Infrastructure Projects," *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, no. 7, pp. 743–747, 2019.

[121] S. M. E. Sepasgozar, R. Karimi, S. Shirowzhan, M. Mojtahedi, S. Ebrahimzadeh, and D. McCarthy, "Delay Causes and Emerging Digital Tools: A Novel Model of Delay Analysis, Including Integrated Project Delivery and PMBOK," *Buildings*, vol. 9, no. 9, 2019, doi: 10.3390/buildings9090191.

[122] Gunanto and Z. F. Ikatrinasari, "Delay Analysis of Pressure Vessels Fabrication Using Fuzzy FMEA," *Int. J. Mech. Prod. Eng. Res. Dev.*, vol. 9, no. 2, pp. 83–90, 2019, doi: 10.24247/ijmperdapr20198.

[123] S. Makesh and M. Mathivanan, "Analysis on Causes of Delay in Building Construction," *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, no. 7, pp. 335–341, 2019.

[124] N. W. Shengea, R. N. Misra, and S. K. Mishra, "Construction Delay Analysis of Some indian Hydropower Projects," *Water Energy Int.*, vol. 62r, no. 10, pp. 61–67, 2020.

[125] S. Hussain, F. Zhu, Z. Ali, H. D. Aslam, and A. Hussain, "Critical Delaying Factors: Public Sector Building Projects in Gilgit-Baltistan, Pakistan," *Buildings*, vol. 8, no. 1, pp. 1–16, 2018, doi: 10.3390/buildings8010006.

[126] A. Shebob, N. Dawood, and Q. Xu, "Analysing Construction Delay Factors: A Case Study of Building Construction Project in Libya," in *Association of Researchers in Construction Management, ARCOM 2011 - Proceedings of the 27th Annual Conference*, 2011, vol. 2, pp. 1005–1012.

[127] A. Shebob, N. Dawood, R. K. Shah, and Q. Xu, "Comparative Study of Delay Factors in Libyan and the UK Construction Industry," *Eng. Constr. Archit. Manag.*, vol. 19, no. 6, pp. 688–712, 2012, doi: 10.1108/09699981211277577.

[128] D. Y. Liu and J. G. Chen, *Construction Delay Analysis and Early Warning Model Based on Design Structure Matrix*, vol. 357–360. 2013.

[129] T. Ramachandra, J. O. Rotimi, and S. Gunaratne, "Reasons for Contractors' Delay Claims Failures in Sri Lanka," in *Proceedings 30th Annual Association of Researchers in Construction Management Conference, ARCOM 2014*, 2014, pp. 475–484.

[130] C. Manoj Kumaar and T. R. Praveen Kumar, "A Dynamic Scheduling Technique to Control Time in Construction Projects," *Int. J. Appl. Eng. Res.*, vol. 10, no. 62, pp. 338–344, 2015.

[131] M. Ghanbari, M. Hosseinalipour, and S. F. Mousavi, "Two Proposed Basis Strategies for Delay Claims Assessments in Engineering Projects: An Iranian

Perspective," in *ICCREM 2016: BIM Application and Offsite Construction - Proceedings of the 2016 International Conference on Construction and Real Estate Management*, 2016, pp. 932–939, doi: 10.1061/9780784480274.113.

[132] M. P. Jalal and S. Shoar, "A Hybrid SD-Dematel Approach to Develop a Delay Model for Construction Projects," *Eng. Constr. Archit. Manag.*, vol. 24, no. 4, pp. 629–651, 2017, doi: 10.1108/ECAM-02-2016-0056.

[133] K. Adiam and H. Nel, "Delay Analysis in Energy Utility Maintenance Project," in *Proceedings of the International Conference on Industrial Engineering and Operations Management*, 2019, no. July, pp. 343–354.

[134] J.-S. Lee and J. E. Diekmann, "Delay Analysis Considering Production Rate," *Can. J. Civ. Eng.*, vol. 38, no. 2, pp. 361–372, 2011.

[135] S. Golanaraghi and S. Alkass, "Modified Isolated Delay Type Technique," in *Construction Research Congress 2012: Construction Challenges in a Flat World, Proceedings of the 2012 Construction Research Congress*, 2012, pp. 90–99, doi: 10.1061/9780784412329.010.

[136] N. Braimah, "Construction Delay Analysis Techniques—A Review of Application Issues and Improvement Needs," *Buildings*, vol. 3, no. 3, pp. 506–531, 2013, doi: 10.3390/buildings3030506.

[137] K. S. Al-Gahtani, I. A. Al-Sulaihi, and A. Iqupal, "Total Float Management: Computerized Technique for Construction Delay Analysis," *Can. J. Civ. Eng.*, vol. 43, no. 5, pp. 391–401, 2016, doi: 10.1139/cjce-2015-0434.

[138] Z. R. Kamandang, J.-B. Yang, and I. Wijatmiko, "An Empirical Study on Analyzing Schedule Delays of Construction Project in Indonesia," in *ISARC 2017 - Proceedings of the 34th International Symposium on Automation and Robotics in Construction*, 2017, pp. 544–551.

[139] M. Bhih and T. Hegazy, "Concurrent Delays: Comparison Among Forensic Analysis Recommended Practices," in *Proceedings, Annual Conference - Canadian Society for Civil Engineering*, 2019, vol. 2019-June.

[140] M. Bhih and T. Hegazy, "Improving Concurrency Assessment and Resolving Misconceptions about But-For Delay Analysis Technique," *J. Leg. Aff. Disput. Resolut. Eng. Constr.*, vol. 12, no. 2, pp. 04520007-1–10, 2020, doi: 10.1061/(ASCE)LA.1943-4170.0000378.

[141] C. Munvar, D. G. Mengistu, and G. Mahesh, "Concurrent Delay Analysis: Methods, Case Law, and Expert Perception," *J. Leg. Aff. Disput. Resolut. Eng. Constr.*, vol. 12, no. 1, 2020, doi: 10.1061/(ASCE)LA.1943-4170.0000343.

[142] P. J. Keane and A. F. Caletka, *Delay Analysis in Construction Contracts*, 2. 2015.

[143] T. T. Calvey and R. M. Winter, "AACE International Recommended Practice No. 52R-06," 2006.

[144] C. Budayan, O. Okudan, and I. Dikmen, “Identification and Prioritization of Stage-Level KPIs for BOT Projects – Evidence from Turkey,” *Int. J. Manag. Proj. Bus.*, vol. 13, no. 6, pp. 1311–1337, Jul. 2020, doi: 10.1108/IJMPB-11-2019-0286.

[145] J. M. De La Garza, A. Prateapusanond, and N. Ambani, “Preallocation of Total Float in the Application of a Critical Path Method Based Construction Contract,” *J. Constr. Eng. Manag.*, vol. 133, no. 11, pp. 83–845, 2007, doi: 10.1061/ASCE0733-93642007133:11836.

[146] E. Bayrak, “Importance of Float Management in Contractor’s Extension of Time Claims, a Case Study,” Istanbul Technical University, 2010.

[147] S. Dere, “İnşaat Projelerinde Gecikme Analizi: Bir Gecikme Analizi Uygulaması,” T.C. İstanbul Kültür Üniversitesi, 2013.

[148] S. Alkass, M. Mazerolle, and F. Harris, “Construction Delay Analysis Techniques,” *Constr. Manag. Econ.*, vol. 14, no. 5, pp. 375–394, 1996, doi: 10.1080/014461996373250.

[149] AACE International, “AACE International Recommended Practice No. 29R-03 Forensic Schedule Analysis,” *AACE International*. 2007.

[150] Y. Shan and P. M. Goodrum, “Integration of Building Information Modeling and Critical Path Method Schedules to Simulate the Impact of Temperature and Humidity at the Project Level,” *Buildings*, vol. 4, no. 3, pp. 295–319, 2014, doi: 10.3390/buildings4030295.

[151] M. Guévremont and A. Hammad, “Visualization of Delay Claim Analysis Using 4D Simulation,” *J. Leg. Aff. Disput. Resolut. Eng. Constr.*, vol. 10, no. 3, pp. 1–8, 2018, doi: 10.1061/(ASCE)LA.1943-4170.0000267.

[152] Y. A. S. Essawy and K. Nassar, “Elemental Graph Data Model: A Semantic and Topological Representation of Building Elements,” *Int. J. Civ. Environ. Eng.*, vol. 11, no. 6, pp. 845–856, 2017.

[153] M. Guévremont and A. Hammad, “Review and Survey of 4D Simulation Applications in Forensic Investigation of Delay Claims in Construction Projects,” *J. Leg. Aff. Disput. Resolut. Eng. Constr.*, vol. 12, no. 3, pp. 04520017-1–9, 2020, doi: 10.1061/(ASCE)LA.1943-4170.0000391.

[154] Z. Soltani, S. Anderson, and J. Kang, “The Challenges of Using BIM in Construction Dispute Resolution Process,” *53rd ASC Annu. Int. Conf. Proc.*, pp. 771–776, 2017.

[155] Elagiry, Marino, Lasarte, Elguezabal, and Messervey, “BIM4Ren: Barriers to BIM Implementation in Renovation Processes in the Italian Market,” *Buildings*, vol. 9, no. 9, pp. 1–17, 2019, doi: https://doi.org/10.3390/buildings9090200.

[156] M. Bhih and T. Hegazy, “Enhanced Daily Windows Delay-Analysis Technique,” *J. Leg. Aff. Disput. Resolut. Eng. Constr.*, vol. 13, no. 3, pp. 1–12, 2021, doi: 10.1061/(asce)la.1943-4170.0000475.

[157] M. Bhih and T. Hegazy, "Multiple-window Modified But-For analysis of Project Delays and Accelerations," *Int. J. Constr. Manag.*, pp. 1–11, 2021, doi: 10.1080/15623599.2021.1916708.

[158] A. Monteiro and J. Poças Martins, "Sigabim: A Framework for BIM Application," *38th IAHS World Congr. Visions Futur. housing, Magacities*, pp. 422–427, 2012.

[159] S. Vass and T. K. Gustavsson, "Challenges When Implementing BIM for Industry Change," *Constr. Manag. Econ.*, vol. 35, no. 10, pp. 597–610, 2017, doi: 10.1080/01446193.2017.1314519.

[160] A. H. A. Jamil and M. S. Fathi, "Contractual Challenges for BIM-based Construction Projects: A Systematic Review," *Built Environ. Proj. Asset Manag.*, vol. 8, no. 4, pp. 372–3875, 2018.

[161] M. Zibert, M. Lah, and S. Samo, "Challenges and Opportunities in Implementing BIM Methodology in Tunnelling," *Geomech. Tunn.*, vol. 11, no. 4, pp. 335–339, 2018.

[162] H. J. Bargstädt, "Challenges of BIM for Construction Site Operations," *Procedia Eng.*, vol. 117, no. 1, pp. 52–59, 2015, doi: 10.1016/j.proeng.2015.08.123.

[163] J. H. Woo, "BIM (Building Information Modeling) and Pedagogical Challenges," *Proc. 43rd ASC Natl. Annu. Conf.*, pp. 12–14, 2006, doi: 10.1038/nnano.2015.159.

[164] M. Sediqi, "4D BIM Adoption: The Incentives for and Barriers to 4D BIM Adoption within Swedish Construction Companies," Royal Institute of Technology Department, 2018.

[165] F. Salah, "Investigation of Strengths and Weaknesses of 4D BIM Software Applications in Managing Construction Projects," University of Gaziantep, 2014.

A

DEVELOPING PSEUDO CODE

```
int checkEq1(taskMS, taskMUS, window){
   if (taskMS has no predecessor
      &&
      taskMS.EarlyStartDate >= window.StartTime
      &&
      &&
      taskMUS.EarlyStartDate >= window.StartTime
      &&
      taskMUS.EarlyStartDate <= window.FinishTime
   ) {
      return taskMS.EarlyStartDate - taskMUS.EarlyStartDate;
   }
   return null;
   }
int checkEq2(taskMS, taskMUS, window){
   if (taskMS has no predecessor
      &&
      taskMS.EarlyStartDate >= window.StartTime
      &&
      taskMS.EarlyStartDate <= window.FinishTime
      &&
```

```
        taskMUS.EarlyStartDate > window.FinishTime
    ) {
        return taskMS.EarlyStartDate - window.FinishTime -1;
    }
    return null;
}
int checkEq3(taskMS, taskMUS, window){
    Pred predecessorMS = findMaxPredecessor(taskMS.predecessors);
    Pred predecessorMUS = findByID(taskMUS, predecessorMS);
    if (drivingPredecessorMS.Type.Equals(TaskRelationType.FinishStart)){
        if(predecessorMS.EarlyFinishDate >= window.StartTime &&
predecessorMS.EarlyFinishDate <= window.FinishTime
              &&
            taskMUS.EarlyStartDate >= window.Start && taskMUS.EarlyStartDate
<= window.Finish
             &&
            predecessorMUS.EarlyFinishDate >= window.StartTime &&
predecessorMUS.EarlyFinishDate <= window.FinishTime
              ){
            return (predecessorMUS.EarlyFinishDate - taskMUS.EarlyStartDate +
(int)drivingPredecessorMUS.Lag)
        }
    }
    if (taskMS.EarlyStartDate >= window.StartTime
            &&
```

```
        drivingPredecessorMS.Type.Equals(TaskRelationType.FinishStart)
          &&
        predecessorMS.EarlyFinishDate < window.StartTime
          &&
        (int)drivingPredecessorMS.Lag > ( window.StartTime -
predecessorMUS.EarlyFinishDate )
          &&
        taskMUS.EarlyStartDate >= window.StartTime
          &&
        taskMUS.EarlyStartDate <= window.FinishTime
    ) {
        return (predecessorMUS.EarlyFinishDate - taskMUS.EarlyStartDate +
(int)drivingPredecessorMUS.Lag)
    }
    return null;
}
int checkEq4(taskMS, taskMUS, window){
    Pred predecessorMS = findMaxPredecessor(taskMS.predecessors);
    Pred predecessorMUS = findByID(taskMUS, predecessorMS);
    if (taskMS.EarlyStartDate >= window.StartTime
          &&
    if (taskMS.EarlyStartDate <= window.FinishTime
          &&
        drivingPredecessorMS.Type.Equals(TaskRelationType.FinishStart)
          &&
```

```
    predecessorMS.EarlyFinishDate >= window.StartTime
        &&
    predecessorMS.EarlyFinishDate <= window.FinishTime
        &&
    taskMUS.EarlyStartDate >= window.StartTime
        &&
    taskMUS.EarlyStartDate <= window.FinishTime
        &&
    predecessorMUS.EarlyFinishDate > window.FinishTime
) {
        return (taskMUS.EarlyStartDate - window.FinishTime -1);
}
if (
    taskMS.EarlyStartDate >= window.StartTime
        &&
    taskMS.EarlyStartDate <= window.FinishTime
        &&
    drivingPredecessorMS.Type.Equals(TaskRelationType.StartStart)
        &&
    predecessorMS.EarlyStartDate >= window.StartTime
        &&
    predecessorMS.EarlyStartDate <= window.FinishTime
        &&
    taskMUS.EarlyStartDate >= window.StartTime
```

```
        &&
      taskMUS.EarlyStartDate <= window.FinishTime
        &&
      predecessorMUS.EarlyStartDate > windo.finishTime
    ) {
      return (taskMUS.EarlyStartDate - window.FinishTime -1);
    }
    return null;
}
int checkEq5(taskMS, taskMUS, window){
    Pred predecessorMS = findMaxPredecessor(taskMS.predecessors);
    Pred predecessorMUS = findByID(taskMUS, predecessorMS);
    bool condition1 = (
      drivingPredecessorMS.Type.Equals(TaskRelationType.FinishFinish)
        &&
      predecessorMS.EarlyFinishDate >= window.StartTime
        &&
      predecessorMS.EarlyFinishDate <= window.FinishTime
        &&
      taskMS.EarlyFinishDate >= window.StartTime
        &&
      taskMS.EarlyFinishDate <= window.FinishTime
        &&
      predecessorMUS.EarlyFinishDate >= window.StartTime
```

```
        &&
    predecessorMUS.EarlyFinishDate <= window.FinishTime
        &&
    taskMUS.EarlyFinishDate >= window.StartTime
        &&
    taskMUS.EarlyFinishDate <= window.FinishTime
);
bool condition2 =(
    taskMS.EarlyStartDate >= window.StartTime
        &&
    taskMS.EarlyStartDate <= window.FinishTime
        &&
    drivingPredecessorMS.Type.Equals(TaskRelationType.FinishFinish)
        &&
    predecessorMS.EarlyFinishDate < window.StartTime
        &&
    (int)drivingPredecessorMUS.Lag > ( window.StartTime -
predecessorMUS.EarlyFinishDate )
        &&
    taskMUS.EarlyStartDate >= window.StartTime
        &&
    taskMUS.EarlyStartDate <= window.FinishTime
);
if (condition1 || condition2)
{
```

```
        return (predecessorMUS.EarlyFinishDate - taskMUS.EarlyFinishDate
+(int)drivingPredecessorMUS.Lag);
    }
    return null;
}
int checkEq6(taskMS, taskMUS, window){
    Pred predecessorMS = findMaxPredecessor(taskMS.predecessors);
    Pred predecessorMUS = findByID(taskMUS, predecessorMS);
    if( taskMS.EarlyFinishDate >= window.StartTime
            &&
        taskMS.EarlyFinishDate <= window.FinishTime
            &&
        drivingPredecessorMS.Type.Equals(TaskRelationType.FinishFinish)
            &&
        predecessorMS.EarlyFinishDate >= window.StartTime
            &&
        predecessorMS.EarlyFinishDate <= window.FinishTime
            &&
        predecessorMUS.EarlyFinishDate > window.FinishTime
            &&
        taskMUS.EarlyFinishDate >= window.StartTime
            &&
        taskMUS.EarlyFinishDate <= window.FinishTime
    ) {
        return (taskMUS.EarlyFinishDate - window.FinishTime)
```

```
    }
    return null;
}
int checkEq7(taskMS, taskMUS, window){
    Pred predecessorMS = findMaxPredecessor(taskMS.predecessors);
    Pred predecessorMUS = findByID(taskMUS, predecessorMS);
    bool condition1 = (
        taskMS.EarlyStartDate >= window.StartTime
            &&
        taskMS.EarlyStartDate <= window.FinishTime
            &&
        drivingPredecessorMS.Type.Equals(TaskRelationType.StartStart)
            &&
        predecessorMS.EarlyStartDate >= window.StartTime
            &&
        predecessorMS.EarlyStartDate <= window.FinishTime
            &&
        taskMUS.EarlyStartDate >= window.StartTime
            &&
        taskMUS.EarlyStartDate <= window.FinishTime
            &&
        predecessorMUS.EarlyStartDate >= window.StartTime
            &&
        predecessorMUS.EarlyStartDate <= window.FinishTime
```

```
    );
    bool condition2 = (
        taskMS.EarlyStartDate >= window.StartTime
            &&
        taskMS.EarlyStartDate <= window.FinishTime
            &&
        drivingPredecessorMS.Type.Equals(TaskRelationType.StartStart)
            &&
        predecessorMS.EarlyStartDate <=window.StartTime
            &&
        (int)drivingPredecessorMUS.Lag > ( window.StartTime -
predecessorMUS.EarlyStartDate )
            &&
        taskMUS.EarlyStartDate >= window.StartTime
            &&
        taskMUS.EarlyStartDate <= window.FinishTime
    );
    if (condition1 || condition2) {
        return (predecessorMUS.EarlyStartDate - taskMUS.EarlyStartDate +
(int)drivingPredecessorMUS.Lag);
    }
    return null;
}
int checkEq8(taskMS, taskMUS, window){
    Pred predecessorMS = findMaxPredecessor(taskMS.predecessors);
```

```
    Pred predecessorMUS = findByID(taskMUS, predecessorMS);
    if(
        taskMS.EarlyStartDate >= window.StartTime
            &&
        taskMS.EarlyStartDate <= window.FinishTime
            &&
        drivingPredecessorMS.Type.Equals(TaskRelationType.FinishFinish)
            &&
        predecessorMS.EarlyFinishDate < window.StartTime
            &&
        (int)drivingPredecessorMUS.Lag < ( window.StartTime -
predecessorMUS.EarlyFinishDate )
            &&
        taskMUS.EarlyStartDate >= window.StartTime
            &&
        taskMUS.EarlyFinishDate <= window.FinishTime
    ) {
        return (window.StartTime - taskMUS.EarlyFinishDate - 1);
    }
    return null;
}
int checkEq9(taskMS, taskMUS, window){
    Pred predecessorMS = findMaxPredecessor(taskMS.predecessors);
    Pred predecessorMUS = findByID(taskMUS, predecessorMS);
    bool condition1 = (
```

```
taskMS.EarlyStartDate >= window.StartTime
    &&
taskMS.EarlyStartDate <= window.FinishTime
    &&
drivingPredecessorMS.Type.Equals(TaskRelationType.StartStart)
    &&
predecessorMS.EarlyStartDate < window.StartTime
    &&
(int)drivingPredecessorMUS.Lag < ( window.StartTime -
predecessorMUS.EarlyStartDate )
    &&
taskMUS.EarlyStartDate >= window.StartTime
    &&
taskMUS.EarlyStartDate <= window.FinishTime
);
bool condition2 = (
taskMS.EarlyStartDate >= window.StartTime
    &&
taskMS.EarlyStartDate <= window.FinishTime
    &&
drivingPredecessorMS.Type.Equals(TaskRelationType.FinishStart)
    &&
predecessorMS.EarlyFinishDate < window.StartTime
    &&
```

```
        (int)drivingPredecessorMUS.Lag < window.StartTime -
predecessorMUS.EarlyFinishDate
            &&
        taskMUS.EarlyStartDate >= window.StartTime
            &&
        taskMUS.EarlyStartDate <= window.FinishTime
    );
    if (condition1 || condition2) {
        return (window.StartTime - taskMUS.EarlyStartDate);
    }
    return null;
}
int checkEq10(taskMS, taskMUS, window){
    Pred predecessorMS = findMaxPredecessor(taskMS.predecessors);
    Pred predecessorMUS = findByID(taskMUS, predecessorMS);
    bool condition1 = (
        taskMS.EarlyStartDate >= window.StartTime
            &&
        taskMS.EarlyStartDate <= window.FinishTime
            &&
        drivingPredecessorMS.Type.Equals(TaskRelationType.FinishStart)
            &&
        predecessorMS.EarlyStartDate >= window.StartTime
            &&
        predecessorMS.EarlyStartDate <= window.FinishTime
```

```
        &&
      taskMUS.EarlyStartDate > window.FinishTime
        &&
      predecessorMUS.EarlyFinishDate >= window.StartTime
        &&
      predecessorMUS.EarlyFinishDate <= window.FinishTime
    );
positive values are disregarded
    bool condition2 = (
      taskMS.EarlyStartDate >= window.StartTime
        &&
      taskMS.EarlyStartDate <= window.FinishTime
        &&
      drivingPredecessorMS.Type.Equals(TaskRelationType.FinishFinish)
        &&
      predecessorMS.EarlyFinishDate >= window.StartTime
        &&
      predecessorMS.EarlyFinishDate <= window.FinishTime
        &&
      taskMUS.EarlyStartDate >= window.StartTime
        &&
      taskMUS.EarlyStartDate <= window.FinishTime
        &&
      taskMUS.EarlyFinishDate > window.FinishTime
```

```
        &&
      predecessorMUS.EarlyFinishDate >= window.StartTime
        &&
      predecessorMUS.EarlyFinishDate <= window.FinishTime
  );
positive values are disregarded
  bool condition3 = (
      taskMS.EarlyStartDate >= window.StartTime
        &&
      taskMS.EarlyStartDate <= window.FinishTime
        &&
      drivingPredecessorMS.Type.Equals(TaskRelationType.FinishStart)
        &&
      predecessorMS.EarlyFinishDate < window.FinishTime
        &&
      (int)drivingPredecessorMUS.Lag > ( window.StartTime -
predecessorMUS.EarlyFinishDate )
        &&
      taskMUS.EarlyStartDate > window.FinishTime
  );
positive values are disregarded
  bool condition4 = (
      taskMS.EarlyStartDate >= window.StartTime
        &&
      taskMS.EarlyStartDate <= window.FinishTime
```

```
        &&
    drivingPredecessorMS.Type.Equals(TaskRelationType.FinishFinish)
        &&
    predecessorMS.EarlyFinishDate < window.StartTime
        &&
    (int)drivingPredecessorMUS.Lag > ( window.StartTime -
predecessorMUS.EarlyFinishDate )
        &&
    taskMUS.EarlyFinishDate > window.FinishTime
  );
  if (condition1 || condition2 || condition3 || condition4) {
    return (predecessorMUS.EarlyFinishDate - window.FinishTime +
(int)drivingPredecessorMUS.Lag);
  }
  return null;
}
int checkEq11(taskMS, taskMUS, window){
  Pred predecessorMS = findMaxPredecessor(taskMS.predecessors);
  Pred predecessorMUS = findByID(taskMUS, predecessorMS);
  bool condition = (
    taskMS.EarlyStartDate >= window.StartTime
        &&
    taskMS.EarlyStartDate <= window.FinishTime
        &&
    drivingPredecessorMS.Type.Equals(TaskRelationType.StartStart)
```

```
        &&
      predecessorMS.EarlyStartDate >= window.StartTime
        &&
      predecessorMS.EarlyStartDate <= window.FinishTime
        &&
      taskMUS.EarlyStartDate > window.FinishTime
        &&
      predecessorMUS.EarlyStartDate >= window.StartTime
        &&
      predecessorMUS.EarlyStartDate <= window.FinishTime
   );
positive values are disregarded
   condition = condition || (
      taskMS.EarlyStartDate >= window.StartTime
        &&
      taskMS.EarlyStartDate <= window.FinishTime
        &&
      drivingPredecessorMS.Type.Equals(TaskRelationType.StartStart)
        &&
      predecessorMS.EarlyStartDate < window.StartTime
        &&
      (int)drivingPredecessorMUS.Lag > ( window.StartTime -
predecessorMUS.EarlyStartDate )
        &&
      taskMUS.EarlyStartDate >= window.FinishTime
```

```
    );
    if (condition) {
        return (predecessorMUS.EarlyStartDate - window.FinishTime +
(int)drivingPredecessorMUS.Lag - 1);
    }
    return null;
}
int checkEq12(taskMS, taskMUS, window) {
    Pred predecessorMS = findMaxPredecessor(taskMS.predecessors);
    Pred predecessorMUS = findByID(taskMUS, predecessorMS);
    bool condition = (
        taskMS.EarlyStartDate >= window.StartTime
            &&
        taskMS.EarlyStartDate <= window.FinishTime
            &&
        drivingPredecessorMS.Type.Equals(TaskRelationType.FinishStart)
            &&
        predecessorMS.EarlyFinishDate < window.StartTime
            &&
        (int)drivingPredecessorMUS.Lag < ( window.StartTime -
predecessorMUS.EarlyFinishDate )
            &&
        taskMUS.EarlyStartDate > window.FinishTime
    );
    condition = condition || (
```

```
        taskMS.EarlyStartDate >= window.StartTime
            &&
        taskMS.EarlyStartDate <= window.FinishTime
            &&
        drivingPredecessorMS.Type.Equals(TaskRelationType.StartStart)
            &&
        predecessorMS.EarlyStartDate < window.StartTime
            &&
        (int)drivingPredecessorMUS.Lag < ( window.StartTime - predecessorMUS.EarlyStartDate )
            &&
        taskMUS.EarlyStartDate > window.FinishTime
    );
    condition = condition || (
        taskMS.EarlyStartDate >= window.StartTime
            &&
        taskMS.EarlyStartDate <= window.FinishTime
            &&
        drivingPredecessorMS.Type.Equals(TaskRelationType.FinishFinish)
            &&
        predecessorMS.EarlyFinishDate < window.StartTime
            &&
        (int)drivingPredecessorMUS.Lag < window.StartTime - predecessorMUS.EarlyFinishDate
            &&
```

```
        taskMUS.EarlyFinishDate > window.FinishTime
    );
    if (condition) {
        return ( window.StartTime - window.FinishTime );
    }
    return null;
}
int checkEq13(taskMS, taskMUS, window) {
    Pred predecessorMS = findMaxPredecessor(taskMS.predecessors);
    Pred predecessorMUS = findByID(taskMUS, predecessorMS);
    bool condition1 = (
        taskMUS.EarlyStartDate <= window.FinishTime
            &&
        taskMUS.EarlyFinishDate >= window.StartTime
    );
    bool condition2 = (
        !drivingPredecessorMS.Type.Equals(TaskRelationType.StartStart)
            &&
        taskMUS.EarlyStartDate >= window.StartTime
            &&
        taskMUS.EarlyFinishDate <= window.FinishTime
    );
    if (condition1 && condition2) {
        return (taskMS.duration - ( taskMUS.EarlyFinishDate -
taskMUS.EarlyStartDate +1 ));
```

```
    }
    return null;
}
int checkEq14(taskMS, taskMUS, window) {
    Pred predecessorMS = findMaxPredecessor(taskMS.predecessors);
    Pred predecessorMUS = findByID(taskMUS, predecessorMS);
    bool condition1 = (
        taskMUS.EarlyStartDate <= window.FinishTime
            &&
        taskMUS.EarlyFinishDate >= window.StartTime
    );
    bool condition2 = (
        !drivingPredecessorMS.Type.Equals(TaskRelationType.StartStart)
    );
    bool condition3 = (
        taskMUS.EarlyStartDate < window.StartTime
            &&
        taskMUS.EarlyFinishDate <= window.FinishTime
            &&
        taskMUS.EarlyFinishDate >= window.StartTime
    );
    if (condition1 && condition2 && condition3) {
        return (taskMS.EarlyFinishDate - taskMUS.EarlyFinishDate);
    }
```

```
    return null;
}
int checkEq15(taskMS, taskMUS, window) {
    Pred predecessorMS = findMaxPredecessor(taskMS.predecessors);
    bool condition1 = (
        taskMUS.EarlyStartDate <= window.FinishTime
            &&
        taskMUS.EarlyFinishDate >= window.StartTime
    );
    bool condition2 = (
        !drivingPredecessorMS.Type.Equals(TaskRelationType.StartStart)
    );
    bool condition3 = (
        taskMUS.EarlyStartDate < window.StartTime
            &&
        taskMUS.EarlyFinishDate >= window.StartTime
            &&
        taskMUS.EarlyFinishDate <= window.FinishTime
            &&
        taskMS.EarlyFinishDate > window.FinishTime
    );
    bool condition4 = (
        taskMUS.EarlyStartDate >= window.StartTime
            &&
```

```
        taskMUS.EarlyFinishDate >= window.FinishTime
            &&
        taskMUS.EarlyStartDate > window.FinishTime
positive values are not counted
    );
    if (condition1 && condition2 && ( condition3 || condition4 )) {
        return (taskMS.duration - (window.FinishTime -
taskMUS.EarlyStartDate+1));
    }
    return null;
}
int checkEq16(taskMS, taskMUS, window) {
    Pred predecessorMS = findMaxPredecessor(taskMS.predecessors);
    bool condition1 = (
        taskMUS.EarlyStartDate <= window.FinishTime
            &&
        taskMUS.EarlyFinishDate >= window.StartTime
    );
    bool condition2 = (
        !drivingPredecessorMS.Type.Equals(TaskRelationType.StartStart)
    );
    bool condition3 = (
        taskMUS.EarlyStartDate < window.StartTime
            &&
        taskMUS.EarlyFinishDate > window.FinishTime
```

```
            &&
        taskMS.EarlyFinishDate >= window.StartTime
            &&
        taskMS.EarlyFinishDate <= window.FinishTime
    );
    if(condition1 && condition2 && condition3) {
        return (taskMS.EarlyFinishDate - window.FinishTime)
    }
    return null;
}
int checkEq17(taskMS, taskMUS, taskBase, window, isOwnerActivity) {
    Successor successorMS = findMinSuccessor(taskMS.successor);
    bool condition1 = (
        taskBaseline == null
            &&
        taskMUS.EarlyStartDate <= window.FinishTime
            &&
        taskMUS.EarlyFinishDate >= window.StartTime
    );
    if (condition1) {
        bool isOwnerActivity;
        if (!isOwnerActivity) {
            return;
        }
```

```
        bool condition2 = (
            taskMS.hasNoPredecessor
                &&
            taskMS.EarlyStartDate >= window.StartTime
                &&
            taskMS.EarlyStartDate <= window.FinishTime
                &&
            successorMS.EarlyStartDate > window.FinishTime
        );
        if (condition2) {
            return ( taskMS.EarlyStartDate - window.FinishTime -1 );
        }
    }
}
int checkEq18(taskMS, taskMUS, taskBase, window, isOwnerActivity) {
    Successor successorMS = findMinSuccessor(taskMS.successor);
    bool condition1 = (
        taskBaseline == null
            &&
        taskMUS.EarlyStartDate <= window.FinishTime
            &&
        taskMUS.EarlyFinishDate >= window.StartTime
    );
    if (condition1) {
```

```
        if (!isOwnerActivity) {
            return;
        }
        bool condition2 = (
            taskMS.hasNoPredecessor
                &&
            taskMS.EarlyStartDate >= window.StartTime
                &&
            taskMS.EarlyStartDate <= window.FinishTime
                &&
            successorMS.EarlyStartDate >= window.StartTime
                &&
            successorMS.EarlyStartDate =< window.FinishTime
        );
        if (condition2) {
            return (taskMS.EarlyStartDate - successorMS.EarlyStartDate);
        }
    }
}
int checkEq19(taskMS, taskMUS, taskBase, window, isOwnerActivity) {
    Successor successorMS = findMinSuccessor(taskMS.successor);
    Pred predecessorMS = findMaxPredecessor(taskMS.predecessors);
    bool condition1 = (
        taskBaseline == null
```

```
        &&
    taskMUS.EarlyStartDate <= window.FinishTime
        &&
    taskMUS.EarlyFinishDate >= window.StartTime
);
if (condition1) {
    if (!isOwnerActivity) {
        return;
    }
    bool condition2 = (
        taskMS.EarlyStartDate >= window.StartTime
            &&
        taskMS.EarlyStartDate <= window.FinishTime
            &&
        taskMS.EarlyStartDate <= window.FinishTime
            &&
        predecessorMS.EarlyFinishDate >= window.StartTime
            &&
        predecessorMS.EarlyFinishDate <= window.FinishTime
            &&
        successorMS.EarlyStartDate > window.FinishTime
    );
    bool condition3 = (
        taskMS.EarlyStartDate < window.StartTime
```

```
            &&
          predecessorMS.EarlyFinishDate >= window.StartTime
            &&
          predecessorMS.EarlyFinishDate <= window.FinishTime
            &&
          successorMS.EarlyStartDate > window.FinishTime
        );
        if (condition2 || condition3) {
          return (predecessorMS.EarlyFinishDate - window.FinishTime);
        }
      }
    }
    int checkEq12X(taskMS, taskMUS, taskBase, window, isOwnerActivity) {
      Successor successorMS = findMinSuccessor(taskMS.successor);
      Pred predecessorMS = findMaxPredecessor(taskMS.predecessors);
      bool condition1 = (
        taskBaseline == null
          &&
        taskMUS.EarlyStartDate <= window.FinishTime
          &&
        taskMUS.EarlyFinishDate >= window.StartTime
      );
      if (condition1) {
        if (!isOwnerActivity) {
```

```
    return;
}
bool condition2 = (
    (
        taskMS.EarlyStartDate >= window.StartTime
            &&
        taskMS.EarlyStartDate <= window.FinishTime
    )
        ||
    (
        taskMS.EarlyFinishDate >= window.StartTime
            &&
        taskMS.EarlyFinishDate <= window.FinishTime
    )
        &&
    predecessorMS.EarlyFinishDate < window.StartTime
        &&
    successorMS.EarlyStartDate > window.FinishTime
);
bool condition3 = (
    taskMS.hasNoPredecessor
        &&
    taskMS.EarlyStartDate < window.StartTime
        &&
```

```
            successorMS.EarlyStartDate > window.FinishTime
        );
        if (condition2 || condition3) { // Formula 12
            return (window.StartTime - window.FinishTime);
        }
    }
}
int checkEq20(taskMS, taskMUS, taskBase, window, isOwnerActivity) {
    Successor successorMS = findMinSuccessor(taskMS.successor);
    Pred predecessorMS = findMaxPredecessor(taskMS.predecessors);
    bool condition1 = (
        taskBaseline == null
            &&
        taskMUS.EarlyStartDate <= window.FinishTime
            &&
        taskMUS.EarlyFinishDate >= window.StartTime
    );
    if (condition1) {
        if (!isOwnerActivity) {
            return;
        }
        bool condition2 = (
            taskMS.EarlyStartDate < window.StartTime
                &&
```

```
    predecessorMS.EarlyFinishDate < window.StartTime
        &&
    successorMS.EarlyStartDate >= window.StartTime
        &&
    successorMS.EarlyStartDate <= window.FinishTime
);
bool condition3 = (
    taskMS.EarlyStartDate < window.StartTime
        &&
    taskMS.EarlyFinishDate >= window.StartTime
        &&
    taskMS.EarlyFinishDate <= window.FinishTime
        &&
    predecessorMS.EarlyFinishDate < window.StartTime
        &&
    successorMS.EarlyStartDate >= window.StartTime
        &&
    successorMS.EarlyStartDate <= window.FinishTime
);
bool condition4 = (
    taskMS.EarlyStartDate < window.StartTime
        &&
    taskMS.EarlyFinishDate > window.FinishTime
        &&
```

```
            predecessorMS.EarlyFinishDate < window.StartTime
                &&
            successorMS.EarlyStartDate >= window.StartTime
                &&
            successorMS.EarlyStartDate <= window.FinishTime
        );
        bool condition5 = (
            taskMS.hasNoPredecessor
                &&
            taskMS.EarlyStartDate < window.StartTime
                &&
            successorMS.EarlyStartDate >= window.StartTime
                &&
            successorMS.EarlyStartDate <= window.FinishTime
        );
        if (condition2 || condition3 || condition4 || condition5) {
            return (window.StartTime - successorMS.EarlyStartDate);
        }
    }
}
int checkEq21(taskMS, taskMUS, taskBase, window, isOwnerActivity) {
    Successor successorMS = findMinSuccessor(taskMS.successor);
    Pred predecessorMS = findMaxPredecessor(taskMS.predecessors);
    bool condition1 = (
```

```
    taskBaseline == null
        &&
    taskMUS.EarlyStartDate <= window.FinishTime
        &&
    taskMUS.EarlyFinishDate >= window.StartTime
);
if (condition1) {
    if (!isOwnerActivity) {
        return;
    }
    bool condition2 = (
        taskMS.EarlyStartDate >= window.StartTime
            &&
        taskMS.EarlyStartDate <= window.FinishTime
            &&
        predecessorMS.EarlyFinishDate >= window.StartTime
            &&
        predecessorMS.EarlyFinishDate <= window.FinishTime
            &&
        successorMS.EarlyStartDate >= window.StartTime
            &&
        successorMS.EarlyStartDate <= window.FinishTime
    );
    if (condition2) {
```

```
            return (predecessorMS.EarlyFinishDate - successorMS.EarlyStartDate);
        }
    }
}
```

B

DEVELOPING CORE FUNCTIONALITY

```
using Bexel.Api.Scheduling;
using MSvsMUS.Model;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace MSvsMUS.Controller
{
    public class DelayCalculator
    {
        private readonly double DEFAULT_WORKING_HOURS = 8.0;
        private IEnumerable<CalculationStep> calculationSteps { get; }
        private TaskID taskID { get; }
        private Task taskMS { get; }
        private Task taskMUS { get; }
        private Task taskBase { get; }
        private WindowDefinition window { get; }
        private TaskRelation drivingPredecessorMS { get; set; }
        private TaskRelation drivingPredecessorMUS { get; set; }
        private TaskRelation drivingSuccessorMS { get; set; }
        private TaskRelation drivingSuccessorMUS { get; set; }
```

```
    public DelayCalculator(TaskID taskID, WindowDefinition window, Task taskMS, Task taskMUS, Task taskBaseline)
    {
        this.taskID = taskID;
        this.window = window;
        this.taskMS = taskMS;
        this.taskMUS = taskMUS;
        this.taskBase = taskBaseline;
        setDrivingPredecessor();
        setDrivingSuccessor();
        this.calculationSteps = getCalculationSteps();
    }
    public IEnumerable<CalculationResult> Calculate()
    {
        List<CalculationResult> calculationResults = new List<CalculationResult>();
        foreach (var step in calculationSteps)
        {
            if (step.Condition())
            {
                calculationResults.Add(new CalculationResult
                {
                    EquationIndex = step.ID,
                    Result = step.Equation(),
                    Task = taskMS,
```

```
                Window = window,
                Type = step.Type
            });
        }
    }
    return calculationResults;
}
private IEnumerable<CalculationStep> getCalculationSteps()
{
    List<CalculationStep> calculationSteps = new List<CalculationStep>();
    // Equation 1
    calculationSteps.Add(new CalculationStep
    {
        ID = "1",
        Equation = (() => {
            return (this.taskMUS.EarlyStartDate.Date -
this.taskMS.EarlyStartDate.Date).Days;
        }),
        Condition = () => (
            requireHasNoPredecessorMS()
                &&
            activityStartsWithinTheWindowInMS()
                &&
            activityStartsWithinTheWindowInMUS()
        )
```

```
        });
        // Equation 2
        calculationSteps.Add(new CalculationStep
        {
            ID = "2",
            Equation = () => {
                return                  ((window.EndDate.Date                  -
taskMS.EarlyStartDate.Date).Days);
            },
            Condition = () => (
                requireHasNoPredecessorMS()
                    &&
                activityStartsWithinTheWindowInMS()
                    &&
                // it starts within one of the following windows
                //   this.taskMUS.EarlyStartDate.CompareTo(this.window.EndDate)
>= 0
                (this.taskMUS.EarlyStartDate.Date                  -
this.window.EndDate.Date).Days >= 0
            )
        });
        // Equation 3
        calculationSteps.Add(new CalculationStep
        {
            ID = "3",
```

```
            Equation = () => {
               Task predecessorMUS = drivingPredecessorMUS.Predecessor;
               int lag = getLagDuration(drivingPredecessorMUS);
               return (taskMUS.EarlyStartDate.Date -
predecessorMUS.EarlyFinishDate.Date).Days - lag;
            },
            Condition = () =>
            {
               if (drivingPredecessorMS == null || drivingPredecessorMUS ==
null)
               {
                  return false;
               }
               Task predecessorTaskMS = drivingPredecessorMS.Predecessor;
               Task predecessorTaskMUS = drivingPredecessorMUS.Predecessor;
               bool condition = (

drivingPredecessorMS.Type.Equals(TaskRelationType.FinishStart)
                  &&
                  (predecessorTaskMS.EarlyFinishDate.Date -
window.StartDate.Date).Days >= 0
                  &&
                  (predecessorTaskMS.EarlyFinishDate.Date -
window.EndDate.Date).Days <= 0
                  &&
```

```
            (taskMUS.EarlyStartDate.Date - window.StartDate.Date).Days >= 0
            &&
            (taskMUS.EarlyStartDate.Date - window.EndDate.Date).Days <= 0
            &&
            (predecessorTaskMUS.EarlyFinishDate.Date - window.StartDate.Date).Days >= 0
            &&
            (predecessorTaskMUS.EarlyFinishDate.Date - window.EndDate.Date).Days <= 0
            &&
            (taskMS.EarlyStartDate.Date - window.StartDate.Date).Days > 0
            &&
            (taskMS.EarlyStartDate.Date - window.EndDate.Date).Days < 0
        );
        bool condition2 = (
            (taskMS.EarlyStartDate.Date - window.StartDate.Date).Days >= 0
            &&
            (drivingPredecessorMS != null && drivingPredecessorMS.Type.Equals(TaskRelationType.FinishStart))
            &&
            (predecessorTaskMUS.EarlyFinishDate.Date - window.StartDate.Date).Days <= 0
            &&
```

```
            getLagDuration(drivingPredecessorMS)                    >
(window.StartDate.Date - predecessorTaskMUS.EarlyFinishDate.Date).Days
            &&
            (taskMUS.EarlyStartDate.Date   -   window.StartDate.Date).Days
>= 0
            &&
            (taskMUS.EarlyStartDate.Date - window.EndDate.Date).Days <=
0
            &&
            (taskMS.EarlyStartDate.Date - window.StartDate.Date).Days > 0
            &&
            (taskMS.EarlyStartDate.Date - window.EndDate.Date).Days < 0
          );
          return condition || condition2;
        }
      });
      // Equation 4
      calculationSteps.Add(new CalculationStep
      {
        ID = "4",
        Equation = () =>
        {
          return (taskMS.EarlyStartDate.Date - window.EndDate.Date).Days;
        },
        Condition = () =>
```

```
            {
               if (drivingPredecessorMS == null || drivingPredecessorMUS ==
null)
               {
                  return false;
               }
               Task predecessorMS = drivingPredecessorMS.Predecessor;
               Task predecessorMUS = drivingPredecessorMUS.Predecessor;
               bool condition = ((taskMS.EarlyStartDate.Date -
window.StartDate.Date).Days >= 0
                  &&

drivingPredecessorMS.Type.Equals(TaskRelationType.FinishStart)
                  &&
                  (predecessorMS.EarlyFinishDate.Date -
window.StartDate.Date).Days >= 0
                  &&
                  (predecessorMS.EarlyFinishDate.Date -
window.EndDate.Date).Days <= 0
                  &&
                  (taskMUS.EarlyStartDate.Date - window.StartDate.Date).Days
>= 0
                  &&
                  (taskMUS.EarlyStartDate.Date - window.EndDate.Date).Days <=
0
                  &&
```

```
                (predecessorMUS.EarlyFinishDate.Date - window.EndDate.Date).Days >= 0
            );
            condition = condition || (
                (taskMS.EarlyStartDate.Date - window.StartDate.Date).Days >= 0
                    &&
                drivingPredecessorMS.Type.Equals(TaskRelationType.StartStart)
                    &&
                (predecessorMS.EarlyStartDate.Date - window.StartDate.Date).Days >= 0
                    &&
                (predecessorMS.EarlyFinishDate.Date - window.EndDate.Date).Days <= 0
                    &&
                (taskMUS.EarlyStartDate.Date - window.StartDate.Date).Days >= 0
                    &&
                (taskMUS.EarlyStartDate.Date - window.EndDate.Date).Days <= 0
                    &&
                (predecessorMUS.EarlyFinishDate.Date - window.EndDate.Date).Days >= 0
            );
            return condition;
        }
```

```
        });
        // Equation 5
        calculationSteps.Add(new CalculationStep
        {
            ID = "5",
            Equation = () => {
                Task predecessorMUS = drivingPredecessorMUS.Predecessor;
                return ((taskMUS.EarlyFinishDate.Date - predecessorMUS.EarlyFinishDate.Date).Days - getLagDuration(drivingPredecessorMUS));
            },
            Condition = () => {
                if (drivingPredecessorMS == null || drivingPredecessorMUS == null)
                {
                    return false;
                }
                Task predecessorMS = drivingPredecessorMS.Predecessor;
                Task predecessorMUS = drivingPredecessorMUS.Predecessor;
                bool condition = (
                        drivingPredecessorMS.Type.Equals(TaskRelationType.FinishFinish)
                        &&
                    (predecessorMS.EarlyFinishDate.Date - window.StartDate.Date).Days >= 0
                        &&
```

```
(predecessorMS.EarlyFinishDate.Date -
window.EndDate.Date).Days <= 0
    &&
(taskMS.EarlyFinishDate.Date - window.StartDate.Date).Days >=
0
    &&
(taskMS.EarlyFinishDate.Date - window.EndDate.Date).Days <=
0
    &&
(predecessorMUS.EarlyFinishDate.Date -
window.StartDate.Date).Days >= 0
    &&
(predecessorMUS.EarlyFinishDate.Date -
window.EndDate.Date).Days <= 0
    &&
(taskMUS.EarlyFinishDate.Date - window.StartDate.Date).Days
>= 0
    &&
(taskMUS.EarlyFinishDate.Date - window.EndDate.Date).Days
<= 0
);
condition = condition || (
(taskMS.EarlyStartDate.Date - window.StartDate.Date).Days >=
0
    &&
```

```
drivingPredecessorMS.Type.Equals(TaskRelationType.FinishFinish)
                    &&
                  (predecessorMS.EarlyFinishDate.Date                        -
window.StartDate.Date).Days <= 0
                    &&
                  getLagDuration(drivingPredecessorMS)                    >=
(window.StartDate.Date - predecessorMS.EarlyFinishDate.Date).Days
                    &&
                  (taskMUS.EarlyStartDate.Date    -    window.StartDate.Date).Days
>= 0
                    &&
                  (taskMUS.EarlyStartDate.Date - window.EndDate.Date).Days <=
0
                );
                return condition;
            }
        });
        // Equation 6
        calculationSteps.Add(new CalculationStep
        {
            ID = "6",
            Equation = () =>
            {
                return              (taskMUS.EarlyFinishDate.Date              -
window.EndDate.Date).Days;
```

```
            },
            Condition = () =>
            {
               if (drivingPredecessorMS == null || drivingPredecessorMUS ==
null)
               {
                  return false;
               }
               Task predecessorMS = drivingPredecessorMS.Predecessor;
               Task predecessorMUS = drivingPredecessorMUS.Predecessor;
               bool condition = (
                  (taskMS.EarlyFinishDate.Date - window.StartDate.Date).Days >=
0
                     &&
                  ( taskMS.EarlyFinishDate.Date - window.EndDate.Date ).Days <=
0
                     &&

drivingPredecessorMS.Type.Equals(TaskRelationType.FinishFinish)
                     &&
                  ( predecessorMS.EarlyFinishDate.Date - window.StartDate.Date
).Days >= 0
                     &&
                  ( predecessorMS.EarlyFinishDate.Date - window.EndDate.Date
).Days <= 0
                     &&
```

```
            ( predecessorMUS.EarlyFinishDate.Date - window.EndDate.Date
).Days >= 0
                &&
            ( taskMUS.EarlyFinishDate.Date - window.StartDate.Date ).Days
>= 0
                &&
            ( taskMUS.EarlyFinishDate.Date - window.EndDate.Date ).Days
<= 0
          );
          return condition;
        }
      });
      // Equation 7
      calculationSteps.Add(new CalculationStep
      {
        ID = "7",
        Equation = () =>
        {
          Task predecessorMUS = drivingPredecessorMUS.Predecessor;
          return          ((taskMUS.EarlyStartDate.Date          -
predecessorMUS.EarlyStartDate.Date).Days                        -
getLagDuration(drivingPredecessorMUS));
        },
        Condition = () =>
        {
```

```
            if (drivingPredecessorMS == null || drivingPredecessorMUS ==
null)
            {
              return false;
            }
            Task predecessorMS = drivingPredecessorMS.Predecessor;
            Task predecessorMUS = drivingPredecessorMUS.Predecessor;
            bool condition = (
              ( taskMS.EarlyStartDate.Date - window.StartDate.Date ).Days >=
0
                &&
              drivingPredecessorMS.Type.Equals(TaskRelationType.StartStart)
                &&
              ( predecessorMS.EarlyStartDate.Date - window.StartDate.Date
).Days >= 0
                &&
              ( predecessorMS.EarlyStartDate.Date - window.EndDate.Date
).Days <= 0
                &&
              ( taskMUS.EarlyStartDate.Date - window.StartDate.Date ).Days
>= 0
                &&
              ( taskMUS.EarlyStartDate.Date - window.EndDate.Date ).Days
<= 0
            );
            condition = condition || (
```

```
            ( taskMS.EarlyStartDate.Date - window.StartDate.Date ).Days >=
0
              &&
            drivingPredecessorMS.Type.Equals(TaskRelationType.StartStart)
              &&
            ( predecessorMS.EarlyStartDate.Date - window.StartDate.Date
).Days <= 0
              &&
            getLagDuration(drivingPredecessorMS) >=
(window.StartDate.Date - predecessorMUS.EarlyStartDate.Date).Days
              &&
            ( taskMUS.EarlyStartDate.Date - window.StartDate.Date ).Days
>= 0
              &&
            ( taskMUS.EarlyStartDate.Date - window.EndDate.Date ).Days
<= 0
          );
          return condition;
        }
      });
      // Equation 8
      calculationSteps.Add(new CalculationStep
      {
        ID = "8",
        Equation = () =>
        {
```

```
        return (window.StartDate.Date - taskMUS.EarlyFinishDate.Date).Days;
    },
    Condition = () =>
    {
        if (drivingPredecessorMS == null || drivingPredecessorMUS == null)
        {
            return false;
        }
        Task predecessorMS = drivingPredecessorMS.Predecessor;
        Task predecessorMUS = drivingPredecessorMUS.Predecessor;
        bool condition = (
            ( taskMS.EarlyStartDate.Date - window.StartDate.Date ).Days >= 0
                &&
drivingPredecessorMS.Type.Equals(TaskRelationType.FinishFinish)
                &&
            ( predecessorMS.EarlyFinishDate.Date - window.StartDate.Date ).Days <= 0
                &&
            getLagDuration(drivingPredecessorMUS) <= (window.StartDate.Date - predecessorMUS.EarlyFinishDate.Date).Days
                &&
```

```
            ( taskMUS.EarlyStartDate.Date - window.StartDate.Date ).Days
>= 0
             &&
            ( taskMUS.EarlyFinishDate.Date - window.EndDate.Date ).Days
<= 0
          );
          return condition;
        }
      });
      // Equation 9
      calculationSteps.Add(new CalculationStep
      {
        ID = "9",
        Equation = () =>
        {
          return (window.StartDate.Date -
taskMUS.EarlyStartDate.Date).Days;
        },
        Condition = () =>
        {
          if (drivingPredecessorMS == null || drivingPredecessorMUS ==
null)
          {
            return false;
          }
```

```
                Task predecessorMS = drivingPredecessorMS.Predecessor;
                Task predecessorMUS = drivingPredecessorMUS.Predecessor;
                bool condition1 = (
                   ( taskMS.EarlyStartDate.Date - window.StartDate.Date ).Days >=
0
                      &&
                   drivingPredecessorMS.Type.Equals(TaskRelationType.StartStart)
                      &&
                   (   predecessorMS.EarlyStartDate.Date   -   window.StartDate.Date
).Days <= 0
                      &&
                   getLagDuration(drivingPredecessorMUS)                          <=
(window.StartDate.Date - predecessorMUS.EarlyStartDate.Date).Days
                      &&
                   (  taskMUS.EarlyStartDate.Date  -  window.StartDate.Date  ).Days
>= 0
                      &&
                   (  taskMUS.EarlyStartDate.Date  -  window.EndDate.Date  ).Days
<= 0
                );
                bool condition2 = (
                   ( taskMS.EarlyStartDate.Date - window.StartDate.Date ).Days >=
0
                      &&

drivingPredecessorMS.Type.Equals(TaskRelationType.FinishStart)
```

```
                        &&
                        ( predecessorMS.EarlyFinishDate.Date - window.StartDate.Date
).Days <= 0
                        &&
                        getLagDuration(drivingPredecessorMUS)                    <=
(window.StartDate.Date - predecessorMUS.EarlyFinishDate.Date).Days
                        &&
                        ( taskMUS.EarlyStartDate.Date - window.StartDate.Date ).Days
>= 0
                        &&
                        ( taskMUS.EarlyStartDate.Date - window.EndDate.Date ).Days
<= 0
                    );
                    return condition1 || condition2;
                }
            });
            // Equation 10
            calculationSteps.Add(new CalculationStep
            {
                ID = "10",
                Equation = () =>
                {
                    Task predecessorMUS = drivingPredecessorMUS.Predecessor;
                    return                    (window.EndDate.Date                   -
predecessorMUS.EarlyFinishDate.Date).Days                                          -
getLagDuration(drivingPredecessorMUS);
```

```
            },
            Condition = () =>
            {
               if (drivingPredecessorMS == null || drivingPredecessorMUS ==
null)
               {
                  return false;
               }
               Task predecessorMS = drivingPredecessorMS.Predecessor;
               Task predecessorMUS = drivingPredecessorMUS.Predecessor;
               bool condition1 = (
                  ( taskMS.EarlyStartDate.Date - window.StartDate.Date ).Days >=
0
                     &&

drivingPredecessorMS.Type.Equals(TaskRelationType.FinishStart)
                     &&
                  ( predecessorMS.EarlyStartDate.Date - window.StartDate.Date
).Days >= 0
                     &&
                  ( predecessorMS.EarlyStartDate.Date - window.EndDate.Date
).Days <= 0
                     &&
                  ( taskMUS.EarlyStartDate.Date - window.EndDate.Date ).Days
>= 0
                     &&
```

```
            ( predecessorMUS.EarlyFinishDate.Date - window.StartDate.Date ).Days >= 0
             &&
            ( predecessorMUS.EarlyFinishDate.Date - window.EndDate.Date ).Days <= 0
        );
        bool condition2 = (
            ( taskMS.EarlyStartDate.Date - window.StartDate.Date ).Days >= 0
             &&
drivingPredecessorMS.Type.Equals(TaskRelationType.FinishFinish)
             &&
            ( predecessorMS.EarlyFinishDate.Date - window.StartDate.Date ).Days >= 0
             &&
            ( predecessorMS.EarlyFinishDate.Date - window.EndDate.Date ).Days <= 0
             &&
            ( taskMUS.EarlyStartDate.Date - window.StartDate.Date ).Days >= 0
             &&
            ( taskMUS.EarlyStartDate.Date - window.EndDate.Date ).Days <= 0
             &&
```

```
                    ( taskMUS.EarlyFinishDate.Date - window.EndDate.Date ).Days
>= 0
                      &&
                    ( predecessorMUS.EarlyFinishDate.Date - window.StartDate.Date
).Days >= 0
                      &&
                    ( predecessorMUS.EarlyFinishDate.Date - window.EndDate.Date
).Days <= 0
                  );
                  bool condition3 = (
                    ( taskMS.EarlyStartDate.Date - window.StartDate.Date ).Days >=
0
                      &&

drivingPredecessorMS.Type.Equals(TaskRelationType.FinishStart)
                      &&
                    ( predecessorMS.EarlyFinishDate.Date - window.EndDate.Date
).Days <= 0
                      &&
                    getLagDuration(drivingPredecessorMUS)                    >=
(window.StartDate.Date - predecessorMUS.EarlyFinishDate.Date).Days
                      &&
                    ( taskMUS.EarlyStartDate.Date - window.EndDate.Date ).Days
>= 0
                  );
                  bool condition4 = (
```

```
                ( taskMS.EarlyStartDate.Date - window.StartDate.Date ).Days >=
0
                  &&

drivingPredecessorMS.Type.Equals(TaskRelationType.FinishFinish)
                  &&
                (  predecessorMS.EarlyFinishDate.Date  -  window.StartDate.Date
).Days <= 0
                  &&
                getLagDuration(drivingPredecessorMUS)                    >=
(window.StartDate.Date - predecessorMUS.EarlyFinishDate.Date).Days
                  &&
                (  taskMUS.EarlyFinishDate.Date  -  window.EndDate.Date  ).Days
>= 0
              );
              return condition1 || condition2 || condition3 || condition4;
            }
          });
          //Equation 11
          calculationSteps.Add(new CalculationStep
          {
            ID = "11",
            Equation = () =>
            {
              Task predecessorMUS = drivingPredecessorMUS.Predecessor;
```

```
                return (window.EndDate.Date - predecessorMUS.EarlyStartDate.Date).Days - getLagDuration(drivingPredecessorMUS);
            },
            Condition = () =>
            {
                if (drivingPredecessorMS == null || drivingPredecessorMUS == null)
                {
                    return false;
                }
                Task predecessorMS = drivingPredecessorMS.Predecessor;
                Task predecessorMUS = drivingPredecessorMUS.Predecessor;
                bool condition = (
                    ( taskMS.EarlyStartDate.Date - window.StartDate.Date ).Days >= 0
                        &&
                    drivingPredecessorMS.Type.Equals(TaskRelationType.StartStart)
                        &&
                    ( predecessorMS.EarlyStartDate.Date - window.StartDate.Date ).Days >= 0
                        &&
                    ( predecessorMS.EarlyStartDate.Date - window.EndDate.Date ).Days <= 0
                        &&
```

```
        ( taskMUS.EarlyStartDate.Date - window.EndDate.Date ).Days
>= 0
          &&
        ( predecessorMUS.EarlyStartDate.Date - window.StartDate.Date
).Days >= 0
          &&
        ( predecessorMUS.EarlyStartDate.Date - window.EndDate.Date
).Days <= 0
      );
      condition = condition || (
        ( taskMS.EarlyStartDate.Date - window.StartDate.Date ).Days >=
0
          &&
        drivingPredecessorMS.Type.Equals(TaskRelationType.StartStart)
          &&
        ( predecessorMS.EarlyStartDate.Date - window.StartDate.Date
).Days <= 0
          &&
        getLagDuration( drivingPredecessorMUS) >=
(window.StartDate.Date - predecessorMUS.EarlyStartDate.Date).Days
          &&
        ( taskMUS.EarlyStartDate.Date - window.EndDate.Date ).Days
>= 0
      );
      return condition;
    }
```

```
});
// Equation 12
calculationSteps.Add(new CalculationStep
{
   ID = "12",
   Equation = () =>
   {
      return (window.EndDate.Date - window.StartDate.Date).Days;
   },
   Condition = () =>
   {
      if (drivingPredecessorMS == null || drivingPredecessorMUS ==
null)
      {
         return false;
      }
      Task predecessorMS = drivingPredecessorMS.Predecessor;
      Task predecessorMUS = drivingPredecessorMUS.Predecessor;
      bool condition = (
        ( taskMS.EarlyStartDate.Date - window.StartDate.Date ).Days >=
0
           &&
        drivingPredecessorMS.Type.Equals(TaskRelationType.FinishStart)
           &&
```

```
( predecessorMS.EarlyFinishDate.Date - window.StartDate.Date
).Days <= 0
    &&
getLagDuration( drivingPredecessorMUS) <=
(window.StartDate.Date - predecessorMUS.EarlyFinishDate.Date).Days
    &&
( taskMUS.EarlyStartDate.Date - window.EndDate.Date ).Days >=
0
);
condition = condition || (
( taskMS.EarlyStartDate.Date - window.StartDate.Date ).Days >=
0
    &&
drivingPredecessorMS.Type.Equals(TaskRelationType.StartStart)
    &&
( predecessorMS.EarlyStartDate.Date - window.StartDate.Date
).Days <= 0
    &&
getLagDuration( drivingPredecessorMUS) <=
(window.StartDate.Date - predecessorMUS.EarlyStartDate.Date).Days
    &&
( taskMUS.EarlyStartDate.Date - window.EndDate.Date ).Days
>= 0
);
condition = condition || (
```

```
                    ( taskMS.EarlyStartDate.Date - window.StartDate.Date ).Days >=
0
                        &&

drivingPredecessorMS.Type.Equals(TaskRelationType.FinishFinish)
                        &&
                    ( predecessorMS.EarlyFinishDate.Date - window.StartDate.Date
).Days <= 0
                        &&
                    getLagDuration( drivingPredecessorMUS) <=
(window.StartDate.Date - predecessorMUS.EarlyFinishDate.Date).Days
                        &&
                    ( taskMUS.EarlyFinishDate.Date - window.EndDate.Date ).Days
>= 0
                );
                return condition;
            }
        });
        // Equation 13
        calculationSteps.Add(new CalculationStep
        {
            ID = "13",
            Equation = () =>
            {
                int originalDuration = getOriginalDuration();
```

```
return (originalDuration - (taskMUS.EarlyFinishDate.Date -
taskMUS.EarlyStartDate.Date).Days);
},
Condition = () =>
{
    bool condition1 = (
        ( taskMUS.EarlyStartDate.Date - window.EndDate.Date ).Days
<= 0
            &&
        ( taskMUS.EarlyFinishDate.Date - window.StartDate.Date ).Days
>= 0
    );
    bool condition2 = (
        (drivingPredecessorMS == null
            ||

!drivingPredecessorMS.Type.Equals(TaskRelationType.StartStart))
            &&
        //taskMUS.EarlyStartDate.Date >= window.StartDate.Date
        (taskMUS.EarlyStartDate.Date - window.StartDate.Date).Days
>= 0
            &&
        //taskMUS.EarlyFinishDate.Date <= window.EndDate.Date
        (taskMUS.EarlyFinishDate.Date - window.EndDate.Date).Days
<= 0
    );
```

```
            return condition1 && condition2;
        }
    });
    // Equation 14
    calculationSteps.Add(new CalculationStep
    {
        ID = "14",
        Equation = () =>
        {
            return (taskMS.EarlyFinishDate.Date - taskMUS.EarlyFinishDate.Date).Days;
        },
        Condition = () =>
        {
            bool condition1 = (
                (taskMUS.EarlyStartDate.Date - window.StartDate.Date).Days < 0
                    &&
                (taskMUS.EarlyFinishDate.Date - window.EndDate.Date).Days <= 0
            );
            bool condition2 = (
                drivingPredecessorMS == null
                    ||
                !drivingPredecessorMS.Type.Equals(TaskRelationType.StartStart)
```

```
            );
            bool condition3 = (
              (taskMUS.EarlyStartDate.Date - window.StartDate.Date).Days
<= 0
                &&
              (taskMUS.EarlyFinishDate.Date - window.EndDate.Date).Days
<= 0
                &&
              (taskMUS.EarlyFinishDate.Date - window.StartDate.Date).Days
>=0
                &&
              (taskMS.EarlyFinishDate.Date - window.EndDate.Date).Days
<=0
                &&
              (taskMS.EarlyFinishDate.Date - window.StartDate.Date).Days >=
0
            );
            return condition1 && condition2 && condition3;
          }
        });
        // Equation 15
        calculationSteps.Add(new CalculationStep
        {
          ID = "15",
          Equation = () =>
          {
```

```
int originalDuration = getOriginalDuration(); ;
int result = (originalDuration - (window.EndDate.Date - taskMUS.EarlyStartDate.Date).Days);
if (result > 0)
{
   return 0;
}
return result;
},
Condition = () =>
{
   bool condition1 = (
      ( taskMUS.EarlyStartDate.Date - window.StartDate.Date ).Days >= 0
         &&
      ( taskMUS.EarlyStartDate.Date - window.EndDate.Date ).Days <= 0
         &&
      ( taskMUS.EarlyFinishDate.Date - window.EndDate.Date ).Days > 0
   );
   bool condition2 = (
      drivingPredecessorMS == null
         ||
      !drivingPredecessorMS.Type.Equals(TaskRelationType.StartStart)
```

```
            );
            bool condition4 = (
              ( taskMUS.EarlyStartDate.Date - window.StartDate.Date ).Days
>= 0
                &&
              ( taskMUS.EarlyFinishDate.Date - window.EndDate.Date ).Days
>= 0
            );
            return condition1 && condition2 && condition4;
          }
        });
        //Equation 6-x
        calculationSteps.Add(new CalculationStep
        {
          ID = "6x",
          Type = DelayType.DELAYS_OR_IMPROMEVENTS,
          Equation = () =>
          {
            return (taskMUS.EarlyFinishDate.Date -
window.EndDate.Date).Days;
          },
          Condition = () =>
          {
            bool condition3 = (
```

```
            ( taskMUS.EarlyStartDate.Date - window.StartDate.Date ).Days
<= 0
                &&
            ( taskMUS.EarlyFinishDate.Date - window.StartDate.Date ).Days
>= 0
                &&
            ( taskMUS.EarlyFinishDate.Date - window.EndDate.Date ).Days
<= 0
                &&
            ( taskMS.EarlyFinishDate.Date - window.EndDate.Date ).Days >=
0
        );
        return condition3;
    }
});
//Equation 16
calculationSteps.Add(new CalculationStep
{
    ID = "16",
    Equation = () =>
    {
        return (taskMS.EarlyFinishDate.Date - window.EndDate.Date).Days;
    },
    Condition = () =>
    {
```

```
bool condition1 = (
   ( taskMUS.EarlyStartDate.Date - window.EndDate.Date ).Days
<= 0
      &&
   ( taskMUS.EarlyFinishDate.Date - window.StartDate.Date ).Days
>= 0
);
bool condition2 = (
   drivingPredecessorMS == null
      ||
   !drivingPredecessorMS.Type.Equals(TaskRelationType.StartStart)
);
bool condition3 = (
   ( taskMUS.EarlyStartDate.Date - window.StartDate.Date ).Days
<= 0
      &&
   ( taskMUS.EarlyFinishDate.Date - window.EndDate.Date ).Days
>= 0
      &&
   ( taskMS.EarlyFinishDate.Date - window.StartDate.Date ).Days
>= 0
      &&
   ( taskMS.EarlyFinishDate.Date - window.EndDate.Date ).Days <=
0
);
return condition1 && condition2 && condition3;
```

```
    }
});
//Equation 17
calculationSteps.Add(new CalculationStep
{
    ID = "17",
    Equation = () =>
    {
        return (window.EndDate.Date - taskMS.EarlyStartDate.Date).Days;
    },
    Condition = () =>
    {
        // taskbase is not null and
        // taskMUS.EarlyStart <= window.Finish and
        // taskMUS.EarlyFinish >= window.Start
        bool condition1 = (
            taskBase == null
            &&
            ( taskMUS.EarlyStartDate.Date - window.EndDate.Date ).Days
<= 0
            &&
            ( taskMUS.EarlyFinishDate.Date - window.StartDate.Date ).Days
>= 0
        );
        if (condition1)
```

```
                {
                    if (!isOwnerActivity(taskMS))
                    {
                        return false;
                    }
                    if (drivingSuccessorMS == null)
                    {
                        return false;
                    }
                    Task successorMS = drivingSuccessorMS.Successor;
                    bool condition2 = (
                        requireHasNoPredecessorMS()
                        &&
                        ( taskMS.EarlyStartDate.Date - window.StartDate.Date ).Days
>= 0
                        &&
                        ( taskMS.EarlyStartDate.Date - window.EndDate.Date ).Days
<= 0
                        &&
                        ( successorMS.EarlyStartDate.Date - window.EndDate.Date
).Days > 0
                    );
                    return condition2;
                }
                return false;
```

```
        }
    });
    // Equation 18
    calculationSteps.Add(new CalculationStep
    {
        ID = "18",
        Equation = () =>
        {
            Task successorMS = drivingSuccessorMS.Successor;
            return (successorMS.EarlyStartDate.Date - taskMS.EarlyStartDate.Date).Days;
        },
        Condition = () =>
        {
            bool condition1 = (
                taskBase == null
                &&
                ( taskMUS.EarlyStartDate.Date - window.EndDate.Date ).Days <= 0
                &&
                ( taskMUS.EarlyFinishDate.Date - window.StartDate.Date ).Days >= 0
            );
            if (condition1)
            {
```

```
            if (!isOwnerActivity(taskMS))
            {
                return false;
            }
            if (drivingSuccessorMS == null)
                return false;
            Task successorMS = drivingSuccessorMS.Successor;
            bool condition2 = (
                requireHasNoPredecessorMS()
                &&
                ( taskMS.EarlyStartDate.Date - window.StartDate.Date ).Days
>= 0
                &&
                ( taskMS.EarlyStartDate.Date - window.EndDate.Date ).Days
<= 0
                &&
                ( successorMS.EarlyStartDate.Date - window.StartDate.Date
).Days >= 0
                &&
                ( successorMS.EarlyStartDate.Date - window.EndDate.Date
).Days <= 0
            );
            return condition2;
        }
        return false;
```

```
    }
  });
  // Equation 19
  calculationSteps.Add(new CalculationStep
  {
    ID = "19",
    Equation = () =>
    {
      Task predecessorMS = drivingPredecessorMS.Predecessor;
      return (window.EndDate.Date - predecessorMS.EarlyFinishDate.Date).Days;
    },
    Condition = () =>
    {
      bool condition1 = (
        taskBase == null
        &&
        ( taskMUS.EarlyStartDate.Date - window.EndDate.Date ).Days <= 0
        &&
        ( taskMUS.EarlyFinishDate.Date - window.StartDate.Date ).Days >= 0
      );
      if (condition1)
      {
```

```
if (!isOwnerActivity(taskMS))
{
    return false;
}
if (drivingPredecessorMS == null || drivingSuccessorMS ==
null)
    return false;
Task predecessorMS = drivingPredecessorMS.Predecessor;
Task successorMS = drivingSuccessorMS.Successor;
bool condition2 = (
    ( taskMS.EarlyStartDate.Date - window.StartDate.Date ).Days
>= 0
    &&
    ( taskMS.EarlyStartDate.Date - window.EndDate.Date ).Days
<= 0
    &&
    ( predecessorMS.EarlyFinishDate.Date - window.StartDate.Date
).Days >= 0
    &&
    ( predecessorMS.EarlyFinishDate.Date - window.EndDate.Date
).Days <= 0
    &&
    ( successorMS.EarlyStartDate.Date - window.EndDate.Date
).Days > 0
);
bool condition3 = (
```

```
                    ( taskMS.EarlyStartDate.Date - window.StartDate.Date ).Days
< 0
                       &&
                    ( predecessorMS.EarlyFinishDate.Date - window.StartDate.Date
).Days >= 0
                       &&
                    ( predecessorMS.EarlyFinishDate.Date - window.EndDate.Date
).Days <= 0
                       &&
                    ( successorMS.EarlyStartDate.Date - window.EndDate.Date
).Days > 0
                 );
                 return (condition2 || condition3);
              }
              return false;
           }
        });
        //Equation 12X
        calculationSteps.Add(new CalculationStep
        {
           ID = "12X",
           Equation = () =>
           {
              return (window.EndDate.Date - window.StartDate.Date).Days;
           },
```

```
Condition = () =>
{
    bool condition1 = (
        taskBase == null
        &&
        ( taskMUS.EarlyStartDate.Date - window.EndDate.Date ).Days
<= 0
        &&
        ( taskMUS.EarlyFinishDate.Date - window.StartDate.Date ).Days
>= 0
    );
    if (condition1)
    {
        if (!isOwnerActivity(taskMS))
        {
            return false;
        }
        if (drivingPredecessorMS == null || drivingSuccessorMS ==
null)
            return false;
        Task predecessorMS = drivingPredecessorMS.Predecessor;
        Task successorMS = drivingSuccessorMS.Successor;
        bool condition2 = (
            (
```

```
            ( taskMS.EarlyStartDate.Date - window.StartDate.Date
).Days >= 0
              &&
            ( taskMS.EarlyStartDate.Date - window.EndDate.Date ).Days
<= 0
          )
            ||
          (
            ( taskMS.EarlyFinishDate.Date - window.StartDate.Date
).Days >= 0
              &&
            ( taskMS.EarlyFinishDate.Date - window.EndDate.Date
).Days <= 0
          )
            &&
          ( predecessorMS.EarlyFinishDate.Date - window.StartDate.Date
).Days < 0
            &&
          ( successorMS.EarlyStartDate.Date - window.EndDate.Date
).Days > 0
        );
        bool condition3 = (
          requireHasNoPredecessorMS()
            &&
          ( taskMS.EarlyStartDate.Date - window.StartDate.Date ).Days
< 0
```

```
                  &&
                  ( successorMS.EarlyStartDate.Date - window.EndDate.Date
).Days > 0
                );
                return (condition2 || condition3);
            }
            return false;
        }
    });
    // Equation 20
    calculationSteps.Add(new CalculationStep
    {
        ID = "20",
        Equation = () =>
        {
            Task successorMS = drivingSuccessorMS.Successor;
            return (successorMS.EarlyStartDate.Date -
window.StartDate.Date).Days;
        },
        Condition = () =>
        {
            if (drivingPredecessorMS == null || drivingSuccessorMS == null)
                return false;
            Task predecessorMS = drivingPredecessorMS.Predecessor;
            Task successorMS = drivingSuccessorMS.Successor;
```

```
bool condition1 = (
   taskBase == null
      &&
   ( taskMUS.EarlyStartDate.Date - window.EndDate.Date ).Days
<= 0
      &&
   ( taskMUS.EarlyFinishDate.Date - window.StartDate.Date ).Days
>= 0
);
if (condition1)
{
   if (!isOwnerActivity(taskMS))
   {
      return false;
   }
   bool condition2 = (
      ( taskMS.EarlyStartDate.Date - window.StartDate.Date ).Days
< 0
         &&
      ( predecessorMS.EarlyFinishDate.Date - window.StartDate.Date
).Days < 0
         &&
      ( successorMS.EarlyStartDate.Date - window.StartDate.Date
).Days >= 0
         &&
```

```
( successorMS.EarlyStartDate.Date - window.EndDate.Date
).Days <= 0
);
bool condition3 = (
( taskMS.EarlyStartDate.Date - window.StartDate.Date ).Days
< 0
&&
( taskMS.EarlyFinishDate.Date - window.StartDate.Date ).Days
>= 0
&&
( taskMS.EarlyFinishDate.Date - window.EndDate.Date ).Days
<= 0
&&
( predecessorMS.EarlyFinishDate.Date - window.StartDate.Date
).Days < 0
&&
( successorMS.EarlyStartDate.Date - window.StartDate.Date
).Days >= 0
&&
( successorMS.EarlyStartDate.Date - window.EndDate.Date
).Days <= 0
);
bool condition4 = (
( taskMS.EarlyStartDate.Date - window.StartDate.Date ).Days
< 0
&&
```

```
        ( taskMS.EarlyFinishDate.Date - window.EndDate.Date ).Days
> 0
          &&
        ( predecessorMS.EarlyFinishDate.Date - window.StartDate.Date
).Days < 0
          &&
        ( successorMS.EarlyStartDate.Date - window.StartDate.Date
).Days >= 0
          &&
        ( successorMS.EarlyStartDate.Date - window.EndDate.Date
).Days <= 0
      );
      bool condition5 = (
        requireHasNoPredecessorMS()
          &&
        ( taskMS.EarlyStartDate.Date - window.StartDate.Date ).Days
< 0
          &&
        ( successorMS.EarlyStartDate.Date - window.StartDate.Date
).Days >= 0
          &&
        ( successorMS.EarlyStartDate.Date - window.EndDate.Date
).Days <= 0
      );
      return (condition2 || condition3 || condition4 || condition5);
    }
```

```
            return false;
        }
    });
    // Equation 21
    calculationSteps.Add(new CalculationStep
    {
        ID = "21",
        Equation = () =>
        {
            Task successorMS = drivingSuccessorMS.Successor;
            Task predecessorMS = drivingPredecessorMS.Predecessor;
            return (successorMS.EarlyStartDate.Date - predecessorMS.EarlyFinishDate.Date).Days;
        },
        Condition = () =>
        {
            if (drivingSuccessorMS == null || drivingPredecessorMS == null)
                return false;
            Task successorMS = drivingSuccessorMS.Successor;
            Task predecessorMS = drivingPredecessorMS.Predecessor;
            bool condition1 = (
                taskBase == null
                    &&
                ( taskMUS.EarlyStartDate.Date - window.EndDate.Date ).Days <= 0
```

```
        &&
      ( taskMUS.EarlyFinishDate.Date - window.StartDate.Date ).Days
>= 0
    );
    if (condition1)
    {
      if (!isOwnerActivity(taskMS))
      {
        return false;
      }
      bool condition2 = (
        ( taskMS.EarlyStartDate.Date - window.StartDate.Date ).Days
>= 0
          &&
        ( taskMS.EarlyStartDate.Date - window.EndDate.Date ).Days
<= 0
          &&
        ( predecessorMS.EarlyFinishDate.Date - window.StartDate.Date
).Days >= 0
          &&
        ( predecessorMS.EarlyFinishDate.Date - window.EndDate.Date
).Days <= 0
          &&
        ( successorMS.EarlyStartDate.Date - window.StartDate.Date
).Days >= 0
          &&
```

```
                    ( successorMS.EarlyStartDate.Date - window.EndDate.Date ).Days <= 0
                );
                return condition2;
            }
            return false;
        }
    });
    return calculationSteps;
}
private bool requireHasNoPredecessorMS()
{
    TaskRelations relations = this.taskMS.Relations;
    IReadOnlyDictionary<TaskRelationID, TaskRelation> predecessors = relations.Predecessors;
    return predecessors.Count == 0;
}
private bool requireHasNoSuccessorMS()
{
    TaskRelations relations = this.taskMS.Relations;
    IReadOnlyDictionary<TaskRelationID, TaskRelation> successors = relations.Successors;
    return successors.Count == 0;
}
public bool NotCalculate()
```

```
{
   return requireHasNoPredecessorMS() && requireHasNoSuccessorMS();
}
private void setDrivingPredecessor()
{
   this.drivingPredecessorMS = getPredecessor(taskMS);
   this.drivingPredecessorMUS = getPredecessor(taskMUS);
}
private TaskRelation getPredecessor(Task t)
{
      IReadOnlyDictionary<TaskRelationID, TaskRelation> predecessors =
t.Relations.Predecessors;
      KeyValuePair<TaskRelationID, TaskRelation> drivingPredecessor = new
KeyValuePair<TaskRelationID, TaskRelation>();
      DateTime maxFinishDateTime = DateTime.MinValue;
      foreach (var predDict in predecessors)
      {
         TaskRelation predecessor = predDict.Value;
         double lag = getLagDuration(predecessor);
         Task predecessorTask = predecessor.Predecessor;
         DateTime currentPredecessorFinishDate =
getFinishDateOfPredecessor(predecessor);
         if ((currentPredecessorFinishDate - maxFinishDateTime).Days > 0)
         {
            drivingPredecessor = predDict;
```

```
                maxFinishDateTime = currentPredecessorFinishDate;
            }
        }
        return drivingPredecessor.Value;
    }
    private void setDrivingSuccessor()
    {
        this.drivingSuccessorMS = getSuccessor(taskMS);
        this.drivingSuccessorMUS = getSuccessor(taskMUS);
    }
    private TaskRelation getSuccessor(Task t)
    {
        KeyValuePair<TaskRelationID, TaskRelation> drivingSuccessor = new
KeyValuePair<TaskRelationID, TaskRelation>();
        TimeSpan minFinishTimeSpan = TimeSpan.MaxValue;
        IReadOnlyDictionary<TaskRelationID, TaskRelation> successors =
t.Relations.Successors;
        foreach (var sucDict in successors)
        {
            TaskRelation successor = sucDict.Value;
            double lag = getLagDuration(successor);
            TimeSpan currentSuccessorFinishTime =
getFinishDateOfSuccessor(successor);
            if (currentSuccessorFinishTime.CompareTo(minFinishTimeSpan) < 0)
            {
```

```
            drivingSuccessor = sucDict;
            minFinishTimeSpan = currentSuccessorFinishTime;
        }
    }
    return drivingSuccessor.Value;
}
private DateTime getFinishDateOfPredecessor(TaskRelation predecessor)
{
    Task predecessorTask = predecessor.Predecessor;
    double lag = getLagDuration(predecessor);
    if (predecessor.Type.Equals(TaskRelationType.StartStart))
    {
        return predecessorTask.EarlyStartDate.AddDays(lag);
    }
    if (
        predecessor.Type.Equals(TaskRelationType.FinishFinish)
            ||
        predecessor.Type.Equals(TaskRelationType.FinishStart)
    )
    {
        return predecessorTask.EarlyFinishDate.AddDays(lag);
    }
    return predecessorTask.EarlyFinishDate;
}
```

```
    private TimeSpan getFinishDateOfSuccessor(TaskRelation successor)
    {
        Task successorTask = successor.Successor;
        double lag = getLagDuration(successor);
        if (successor.Type.Equals(TaskRelationType.StartStart))
        {
            return
successorTask.EarlyStartDate.AddDays(lag).Subtract(taskMS.EarlyStartDate);
        }
        if (successor.Type.Equals(TaskRelationType.FinishFinish))
        {
            return
successorTask.EarlyFinishDate.AddDays(lag).Subtract(taskMS.EarlyFinishDate);
        }
        // has finish to start relation
        return
successorTask.EarlyStartDate.AddDays(lag).Subtract(taskMS.EarlyFinishDate);
    }
    bool activityStartsWithinTheWindowInMS()
    {
        return activityStartsWithinTheWindow(this.taskMS);
    }
    bool activityStartsWithinTheWindowInMUS()
    {
        return activityStartsWithinTheWindow(this.taskMUS);
```

```
}
private bool activityStartsWithinTheWindow(Task t)
{
   return (
      (t.EarlyStartDate - this.window.StartDate).Days >= 0
            &&
      (t.EarlyStartDate - this.window.EndDate).Days <= 0
   );
}
private bool isOwnerActivity(Task t)
{
   return t.Name.EndsWith("#");
}
private int getOriginalDuration()
{
   return (taskMS.EarlyFinishDate - taskMS.EarlyStartDate).Days;
}
private int getLagDuration(TaskRelation relation)
{
   double lag = relation.Lag;
   if (relation.LagType == TaskRelationLagType.WorkingHours)
   {
      return (int)(lag / DEFAULT_WORKING_HOURS);
   }
```

```
            return (int)lag;
        }
    }
}
```

PUBLICATIONS FROM THE THESIS

Conference Papers

1. M. Çevikbaş and Z. Işık, "Determination and Ranking of Obstacles in BIM Implementation with Relative Importance Index," in *6th international Project and Construction Management Conference (IPCMC2020)*, 2020, no. November, pp. 12–14.

Papers

1. M. Çevikbaş and Z. Işık, "An Overarching Review on Delay Analyses in Construction Projects," *Buildings*, vol. 11, no. 3, pp. 1–25, Mar. 2021, doi: 10.3390/buildings11030109.