aace International

AACE International Recommended Practice No. 29R-03

FORENSIC SCHEDULE ANALYSIS

TCM Framework: 6.4 – Forensic Performance Assessment

DRAFT for Public Comment

All Comments due by December 7, 2010

All comments must refer to page and line number.
Comments should be sent to both of the following editors:

khoshino@pcfconsultants.com
john.livengood@arcadis-us.com

CONTENTS

1. Duration & Lag Variance Analysis

3. METHOD IMPLEMENTATION

3.1. Observational / Static / Gross (MIP 3.1)
- A. Description
- B. Common Names
- C. Recommended Source Validation Protocols
- D. Enhanced Source Validation Protocols
- E. Recommended Implementation Protocols
- F. Enhanced Implementation Protocols
  - 1. Daily Delay Measure
- G. Identification of Critical and Near-Critical Paths
- H. Identification & Quantification of Concurrent Delays & Pacing
- I. Determination & Quantification of Excusable and Compensable Delay
  - 1. Excusable & Compensable Delay (ECD)
  - 2. Excusable & Non-Compensable Delay (END)
- J. Identification & Quantification of Mitigation / Constructive Acceleration
- K. Specific Implementation Procedures & Enhancements
- L. Summary of Considerations In Using the Minimum Protocol
- M. Caveats In Using the Minimum Protocol / Conditions Requiring Enhanced Protocols

3.2. Observational / Static / Periodic (MIP 3.2)
- A. Description
- B. Common Names
- C. Recommended Source Validation Protocols
- D. Enhanced Source Validation Protocols
- E. Recommended Implementation Protocols
- F. Enhanced Implementation Protocols
  - 1. Daily Delay Measure
- G. Identification of Critical & Near-Critical Paths
- H. Identification & Quantification of Concurrent Delays & Pacing
- I. Determination & Quantification of Excusable and Compensable Delay
  - 1. Excusable & Compensable Delay (ECD)
  - 2. Excusable & Non-Compensable Delay (END)
- J. Identification & Quantification of Mitigation / Constructive Acceleration
- K. Specific Implementation Procedures & Enhancements
  - 1. Fixed Periods
  - 2. Variable Periods
- L. Summary of Considerations In Using the Minimum Protocol
- M. Caveats In Using the Minimum Protocol / Conditions Requiring Enhanced Protocols

3.3. Observational / Dynamic / Contemporaneous As-Is (MIP 3.3)
- A. Description
- B. Common Names
- C. Recommended Source Validation Protocols
- D. Enhanced Source Validation Protocols

G. Identification of Critical & Near-Critical Paths
H. Identification & Quantification of Concurrent Delays & Pacing
I. Determination & Quantification of Excusable and Compensable Delay
J. Identification & Quantification of Mitigation / Constructive Acceleration
K. Specific Implementation Procedures & Enhancements
    1. Fixed Periods
    2. Variable Periods
    3. All-Periods vs. Grouped-Periods
L. Summary of Considerations In Using the Minimum Protocol
M. Caveats In Using the Minimum Protocol / Conditions Requiring Enhanced Protocols

<u>3.6. Modeled / Additive / Single Base (MIP 3.6)</u>

A. Description
B. Common Names
C. Recommended Source Validation Protocols
D. Enhanced Source Validation Protocols
E. Recommended Implementation Protocols
F. Enhanced Implementation Protocols
G. Identification of Critical & Near-Critical Paths
H. Identification & Quantification of Concurrent Delays & Pacing
I. Determination & Quantification of Excusable and Compensable Delay
    1. Excusable & Compensable Delay (ECD)
    2. Non-Excusable & Non-Compensable Delay (NND)
    3. Excusable & Non-Compensable Delay (END)
J. Identification & Quantification of Mitigation / Constructive Acceleration
K. Specific Implementation Procedures & Enhancements
    1. Global Insertion
    2. Stepped Insertion
L. Summary of Considerations In Using the Minimum Protocol
M. Caveats In Using the Minimum Protocol / Conditions Requiring Enhanced Protocols

<u>3.7. Modeled / Additive / Multiple Base (MIP 3.7)</u>

A. Description
B. Common Names
C. Recommended Source Validation Protocols
D. Enhanced Source Validation Protocols
E. Recommended Implementation Protocols
F. Enhanced Implementation Protocols
G. Identification of Critical & Near-Critical Paths
H. Identification & Quantification of Concurrent Delays & Pacing
I. Determination & Quantification of Excusable and Compensable Delay
    1. Excusable & Compensable Delay (ECD)
    2. Non-Excusable & Non-Compensable Delay (NND)
    3. Excusable & Non-Compensable Delay (END)
J. Identification & Quantification of Mitigation / Constructive Acceleration
K. Specific Implementation Procedures & Enhancements
    1. Fixed Periods

2. Variable Periods
3. Global Insertion
4. Stepped Insertion

L. Summary of Considerations In Using the Minimum Protocol

M. Caveats In Using the Minimum Protocol / Conditions Requiring Enhanced Protocols

<u>3.8. Modeled / Subtractive / Single Simulation (MIP 3.8)</u>

A. Description

B. Common Names

C. Recommended Source Validation Protocols

D. Enhanced Source Validation Protocols

E. Recommended Implementation Protocols

F. Enhanced Implementation Protocols

G. Identification of Critical & Near-Critical Paths

H. Identification & Quantification of Concurrent Delays & Pacing

I. Determination & Quantification of Excusable and Compensable Delay
1. Excusable & Compensable Delay (ECD)
2. Non-Excusable & Non-Compensable Delay (NND)
3. Excusable & Non-Compensable Delay (END)

J. Identification & Quantification of Mitigation / Constructive Acceleration

K. Specific Implementation Procedures & Enhancements
1. Choice of Analysis Periods
   a. Fixed Periods
   b. Variable Periods
   c. Fixed-Periods vs. Variable-Periods
2. Choice of Modeling Increments
   a. Periodic Modeling
   b. Cumulative Modeling
3. Choice of Extraction Models
   a. Global Extraction
   b. Stepped Extraction
   c. Fixed-Periods vs. Variable-Periods
4. Creating a Collapsible As-Built Schedule
5. Identification of the Analogous Critical Path (ACP)

L. Summary of Considerations In Using the Minimum Protocol

M. Caveats In Using the Minimum Protocol / Conditions Requiring Enhanced Protocols

<u>3.9. Modeled / Subtractive / Multiple Base (MIP 3.9)</u>

A. Description

B. Common Names

C. Recommended Source Validation Protocols

D. Enhanced Source Validation Protocols

E. Recommended Implementation Protocols

F. Enhanced Implementation Protocols

G. Identification of Critical & Near-Critical Paths for Each Periodic Update

H. Identification & Quantification of Concurrent Delays & Pacing

I. Determination & Quantification of Excusable and Compensable Delay

1. Excusable & Compensable Delay (ECD)
2. Non-Excusable & Non-Compensable Delay (NND)
3. Excusable & Non-Compensable Delay (END)

J. Identification & Quantification of Mitigation / Constructive Acceleration

K. Specific Implementation Procedures & Enhancements

1. Choice of Extraction Modes
    - a. Global Extraction
    - b. Stepped Extraction
2. Creating a Collapsible As-Built CPM Schedule
3. Identification of the Analogous Critical Path (ACP)

L. Summary of Considerations In Using the Minimum Protocol

M. Caveats In Using the Minimum Protocol / Conditions Requiring Enhanced Protocols

4. ANALYSIS EVALUATION

<u>4.1. Excusability and Compensability of Delay</u>

A. General Rules

B. Accounting for Concurrent Delay

C. Equitable Symmetry of the Concept

<u>4.2. Identification and Quantification of Concurrent Delay</u>

A. Relevance and Application

B. Various Definitions of Concurrency

C. Pre-requisite Findings Concerning Delays Being Evaluated for Concurrency

1. Two or More delays that are unrelated and Independent
2. Two or More Delays that are the Contractual Responsibility of Different Parties
3. The delay must be Involuntary
4. The Delay must be Substantial and Not easily Curable

D. Functional Requirements Establishing Concurrency and the Factors that Influence Findings

1. Literal Concurrency vs. Functional Concurrency
2. Least Float vs. Negative Float
3. Cause of Delay vs. Effect of Delay
4. Frequency, Duration, and Placement of Analysis Intervals
    - a. Frequency & Duration
    - b. Chronological Placement
5. Order of Insertion or Extraction in Stepped Implementation
6. Hindsight vs. ‘Blindsight’

E. Defining the Net Effect of Concurrent Combinations of Delay

F. Pacing

G. Demonstrating Pacing

1. Existence of the Parent Delay
2. Showing of Contemporaneous Ability to Resume Normal Pace
3. Evidence of Contemporaneous Intent

<u>4.3. Critical Path and Float</u>

A. Identifying the Critical Path

1. Critical Path: Longest Path School vs. Total Float Value School

2. Negative Float: Zero Float School vs. Lowest Float Value School

B. Quantifying 'Near-Critical'
1. Duration of Discrete Delay Events
2. Duration of Each Analysis Interval
3. Historical Rate of Float Consumption
4. Amount of Time or Work Remaining on the Project

C. Identifying the As-Built Critical Path

D. Critical Path Manipulation Techniques
1. Resource Leveling & Smoothing
2. Multiple Calendars
3. Precedence Logic / Lead & Lag
4. Start & Finish Constraints
5. Various Calculation Modes
   a. Schedule Calculation
   b. Duration Calculation
6. Use of Data Date
7. Judgment Calls during the Forensic Process

E. Ownership of Float

4.4. Delay Mitigation and Constructive Acceleration

A. Definitions

B. General Considerations
1. Differences among Acceleration, Constructive Acceleration, and Delay Mitigation
2. Acceleration and Compensability
3. Delay Mitigation and Compensability

C. Elements of Constructive Acceleration
1. Contractor Entitlement to an Excusable Delay
2. Contractor Requests and Establishes Entitlement to a Time Extension
3. Owner Failure to Grant a Timely Time Extension
4. Implied Order by the Owner to Complete More Quickly
5. Contractor Notice of Acceleration
6. Proof of Damages

5. CHOOSING A METHOD

5.1 Factor 1: Contractual Requirements

5.2 Factor 2: Purpose of Analysis

5.3 Factor 3: Source Data Availability and Reliability

5.4 Factor 4: Size of the Dispute

5.5 Factor 5: Complexity of the Dispute

5.6 Factor 6: Budget for Forensic Schedule Analysis

5.7 Factor 7: Time Allowed for Forensic Schedule Analysis

5.8 Factor 8: Expertise of the Forensic Schedule Analyst and Resources Available

5.9 Factor 9: Forum for Resolution and Audience

5.10 Factor 10: Legal or Procedural Requirements

5.11 Factor 11: Custom and usage of Methods on the Project or the Case

1. ORGANIZATION AND SCOPE

1.1. Introduction

The purpose of this Association for the Advancement of Cost Engineering International (AACE) Recommended Practice 29R-03, Forensic Schedule Analysis Practice Guide ("RP") is to provide a unifying reference of basic technical principles and guidelines for the application of critical path method (CPM) scheduling in Forensic Schedule Analysis. In providing this reference, the RP will foster competent schedule analysis and furnish the industry as whole with the necessary technical information to categorize and evaluate the varying forensic schedule analysis methods. The RP discusses certain methods of schedule delay analysis, irrespective of whether these methods have been deemed acceptable or unacceptable by Courts or government boards in various countries around the globe.

The RP/FSAPG is not intended to establish a standard of practice, nor is it intended to be a prescriptive document applied without exception. Therefore, a departure from the recommended protocols should not be automatically treated as an error or a deficiency as long as such departure is based on a conscious and sound application of schedule analysis principles. As with any other Recommended Practice, the RP should be used in conjunction with professional judgment and knowledge of the subject matter. While the recommended protocols contained herein are intended to aid the practitioner in creating a competent work product it may, in some cases, require additional or fewer steps.

AACE recognizes that the method(s) of analysis to be utilized in a given situation, and the manner in which a particular methodology might be implemented, are dependent upon the contract, the facts, applicable law, availability and quality of contemporaneous project documentation, and other circumstances particular to a given situation. Therefore, the RP/FSAPG should be read in its entirety and fully understood before applying or using the information for any purpose. This RP is licensed free of charge to the reader on the condition that the reader refrain from using the RP in a manner which is not consistent with its intended use, and the reader does not quote any of the contents in an out-of-context manner. As with any other Recommended Practice published by AACE, this RP is subject to future revisions as new methodologies are identified; new forensic scheduling software is developed; etc.

Forensic[1] scheduling analysis refers to the study and investigation of events using CPM or other recognized schedule calculation methods. It is recognized that such analyses may potentially be used in a legal proceeding. It is the study of how actual events interacted in the context of a complex model for the purpose of understanding the significance of a specific deviation or series of deviations from some baseline model and their role in determining the sequence of tasks within the complex network.

Forensic schedule analysis, like many other technical fields, is both a science and an art. As such, it relies upon professional judgment and expert opinion and usually requires many subjective decisions. One of the most important of these decisions is what technical approach should be used to measure or quantify delay and identify the effected activities in order to focus on causation. Equally important is how the analyst should apply the chosen method. The desired objective of this RP is to reduce the degree of subjectivity involved in the current state of the art. This is with the full awareness that there are certain types of subjectivity that cannot be minimized, let alone eliminated. Professional judgment and expert opinion ultimately rest on subjectivity, but that subjectivity must be based on diligent factual research and analyses whose procedures can be objectified.

[1] The word 'forensic' is defined as: 1. Relating to, used in, or appropriate for courts of law or for public discussion or argumentation. 2. Of, relating to, or used in debate or argument; rhetorical. 3. Relating to the use of science or technology in the investigation and establishment of facts or evidence in a court of law: *a forensic laboratory*.[9]

For these reasons, the RP focuses on minimizing procedural subjectivity. It does this by defining terminology, identifying methodologies currently used by forensic scheduling analysts, classifying them, and setting forth recommended procedural protocols for the use of these techniques. By describing uniform procedures that increase the transparency of the analytical method and the analyst's thought process, the guidelines set forth herein will increase both the accountability and the testability of an opinion and minimize the need to contend with "black-box" or "voodoo" analyses.

Implementation of this RP should result in minimizing disagreements over technical implementation of accepted techniques and allow the providers and consumers of these services to concentrate on resolving disputes based upon substantive, factual and legal issues.

1.2. Basic Premise and Assumptions

a. Forensic scheduling is a technical field that is associated with, but distinct from, project planning and scheduling. It is not just a subset of planning and scheduling.

b. Procedures that may be sufficient for the purpose of project planning, scheduling, and controls may not necessarily be adequate for forensic schedule analysis.

c. It is assumed that this document will be used by practitioners to foster consistency of practice and be used in the spirit of logical and intellectual honesty.

d. All methods are subject to manipulation as they all involve judgment calls by the analyst whether in preparation or in interpretation.

e. No forensic schedule analysis method is exact. The level of accuracy of the answers produced by each method is a function of the quality of the data used therein, the accuracy of the assumptions, and the subjective judgments made by the forensic schedule analyst.

f. Schedules are a project management tool that, in and of themselves, do not demonstrate root causation or responsibility for delays. Legal entitlement to delay damages should be distinct and apart from the forensic schedule analysis methodologies contained in this RP.

1.3. Scope and Focus

The scope and focus of this RP are:

a. This RP covers the technical aspects of forensic schedule analysis methods. It identifies, defines, and describes the usage of various forensic schedule analysis methods in current use. It is not the intent of the RP to exclude or to endorse any method over others. However, it offers caveats and considerations for usage and cites the best current practices and implementation for each method.

b. The focus of this document is on the technical aspects of forensic scheduling as opposed to the legal aspects. This RP is not intended to be a primary resource for legal factors governing claims related to scheduling, delays, and disruption. However, relevant legal principles are discussed to the extent that they would affect the choice of techniques and their relative advantages and disadvantages.

c. Accordingly, the RP primarily focuses on the use of forensic scheduling techniques and methods for factual analysis and quantification as opposed to assignment of delay responsibility. This, however, does not preclude the practitioner from performing the analysis based on certain assumptions regarding liability.

d. This RP is not intended to be a primer on forensic schedule analysis. The reader is assumed to have advanced, hands-on knowledge of all components of CPM analysis and a working experience in a contract claims environment involving delay issues.

e. This RP not intended to be an exhaustive treatment of CPM scheduling techniques. While the RP explains how schedules generated by the planning and scheduling process become the source data for forensic schedule analysis, it is not intended to be a manual for basic scheduling.

f. This RP is not intended to override contract provisions regarding schedule analysis methods or other mutual agreement by the parties to a contract regarding the same. However, this is not an automatic, blanket endorsement of all methods of delay analysis by the mere virtue of their specification in a contract document. It is noted that contractually specified methods often are appropriate for use during the project in a prospective mode but may be inappropriate for retrospective use.

g. It is not the intent of this RP to intentionally contradict or compete with other similar protocols[2]. All efforts should be made by the user to resolve and reconcile apparent contradictions. AACE requests and encourages all users to notify AACE and bring errors, contradictions, and conflict to its attention.

h. This RP deals with CPM-based schedule analysis methods. It is not the intent of the RP to exclude analyses of simple cases where explicit CPM modeling may not be necessary, and mental calculation is adequate for analysis and presentation. The delineation between simple and complex is admittedly blurry and subjective. For this purpose, a 'simple case' is defined as any CPM network model that can be subjected to mental calculation whose reliability cannot be reasonably questioned and allows for effective presentation to lay persons using simple reasoning and intuitive common sense.

i. Finally, the RP is an advisory document to be used in conjunction with professional judgment based on working experience and knowledge of the subject matter. It is not intended to be a prescriptive document that can be applied without exception. When used as intended, this RP will aid the practitioner in creating a competent work product, but some cases require additional steps and some require less. Thus, a departure from the recommended protocols should not be automatically treated as an error or a deficiency as long as such departure is based on a conscious and sound application of schedule analysis principles.

1.4. Taxonomy and Nomenclature

The industry knows the forensic schedule analysis methods and approaches described herein by various common names. Current usage of these names throughout the industry is loose and undisciplined. It is not the intent of this document to enforce more disciplined use of the common names. Instead, the RP will correlate the common names with a taxonomic classification. This taxonomy will allow for the freedom of regional, cultural, and temporal differences in the use of common names for these methods.

[2] The only other similar protocol known at this time is the "Delay & Disruption Protocol" issued in October 2002 by the Society of Construction Law of the United Kingdom[1]. The DDP has a wider scope than this RP.

The RP correlates the common names for the various methods to taxonomic names much like the biosciences use Latin taxonomic terms to correlate regionally diverse common names of plants and animals. This allows the common variations in terminology to coexist with a more objective and uniform language of technical classification. For example, the implementation of MIP 3.6 (aka "TIA") has a bewildering array of regional variations. Not only that, the method undergoes periodic evolutionary changes while maintaining the same name.

By using taxonomic classifications, the RP allows the discussion of the various forensic analysis methods to become more specific and objective. Thus, the RP will not provide a uniform definition for the common names of the various methods, but it will instead describe in detail the taxonomic classification in which they belong. Figure 1 – *Nomenclature Correspondence* shows the commonly associated names for each of the taxonomic classifications.

The RP's taxonomy is a hierarchical classification system of known methods of schedule impact analysis techniques and methods used to analyze how delays and disruptions affect entire CPM networks. For example, methods like the window analysis and collapsed as-built are included in the taxonomy, while procedures such as fragnet modeling, bar charting, and linear graphing are not included. Procedures are tools, not methods, and therefore are not classified under this taxonomy.

The taxonomy is comprised of five layers: timing, basic and specific methods, and the basic and specific implementation of each method. Please refer to Figure 2 – *Taxonomy of Forensic Schedule Analysis* for a graphic representation of the taxonomy. The elements of the diagrams are explained below.

Taxonomy	1	RETROSPECTIVE														
	2	OBSERVATIONAL							MODELED							
	3	Static Logic			Dynamic Logic				Additive				Subtractive			
	4	3.1 Gross	3.2 Periodic		Contemporaneous Updates (3.3 As-Is or 3.4 Split)		3.5 Modified / Reconstructed Updates		3.6 Single Base[2]		3.7 Multi Base[1]		3.8 Single Simulation		3.9 Multi Simulation[1]	
	5		Fixed Periods	Variable Windows	All Periods	Grouped Periods	Fixed Periods	Variable Windows	Global Insertion	Stepped Insertion	Fixed Periods	Variable Windows or Grouped	Global Extraction	Stepped Extraction	Fixed Periods	Stepped Extraction
Common Names		As-Planned vs As-Built	Window Analysis		Contemporaneous Period Analysis, Time Impact Analysis, Window	Contemporaneous Period Analysis, Time Impact Analysis, Window Analysis	Contemporaneous Period Analysis, Time Impact Analysis	Window Analysis, Time Impact Analysis	Impacted As Planned, What-If	Time Impact Analysis, Impacted As-Planned	Time Impact Analysis	Window Analysis, Impacted As-Planned	Collapsed As-Built	Time Impact Analysis, Collapsed As-Built	Time Impact Analysis, Collapsed As-Built	Time Impact Analysis, Window Analysis, Collapsed As-Built

Footnotes

1. Contemporaneous or Modified / Reconstructed
2. The single base can be the original baseline or an update

Figure 1 – Nomenclature Correspondence (see enlarged size figure in Appendix A)

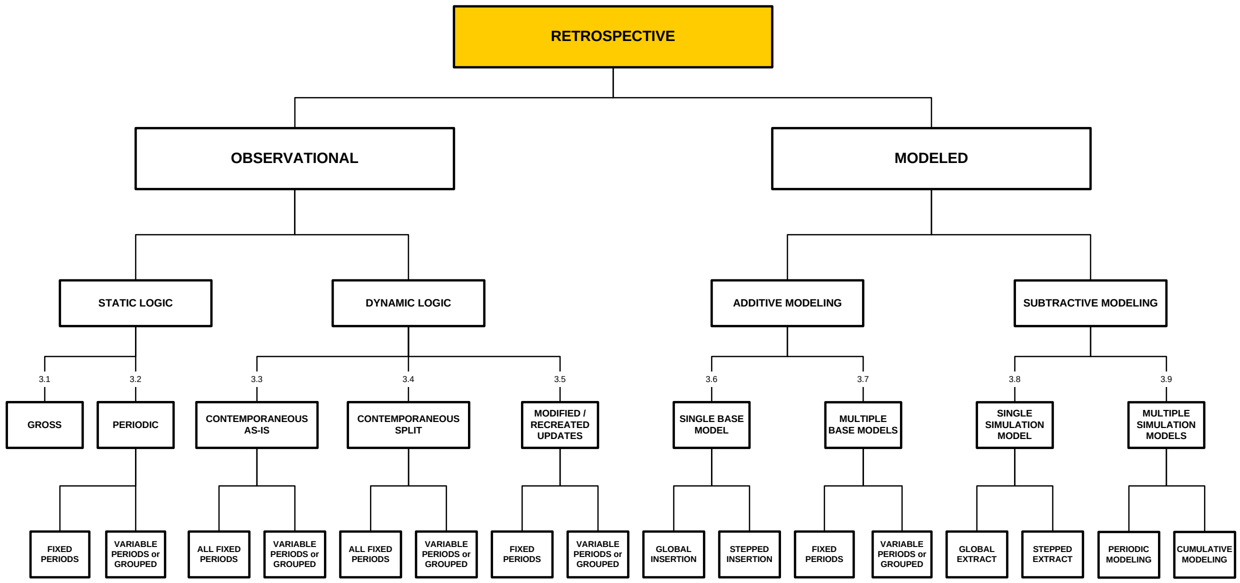


Figure 2 – Taxonomy of Forensic Schedule Analysis (see enlarged size figure in Appendix B)

A. Layer 1: Timing

The first hierarchy layer distinguishes the timing of when the analysis is performed consisting of two branches: prospective and retrospective.

1. Prospective analyses are performed in real-time prior to the delay event or in real-time, contemporaneous with the delay event. In all cases prospective analysis consists of the analyst's best estimate of future events. Prospective analysis occurs while the project is still underway and may not evolve into a forensic context. Since this RP focuses only on Forensic Schedule Analyses, true Prospective schedule analysis methods are not discussed. While some of the methods discussed in this RP employ forward looking calculations they are still performed after the project is completed and are therefore considered Retrospective.

2. Retrospective analyses are performed after the delay event has occurred and the impacts are known. The timing may be soon after the delay event but prior to the completion of the overall project, or after the completion of the entire project. Note that forward-looking analyses (such as 'additive modeling') performed after project completion are still retrospective in terms of timing. What is classified here is the real-time point-of-view of the analyst and not the mode of analysis (forward-looking or hindsight). In other words even forward-looking analysis methods implemented retrospectively have the full benefit of hindsight at the option of the analyst.

This distinction in timing is one of the most significant factors in the choice of methods. For example, contract provisions prescribing methods of delay analysis typically contemplate the preparation of such analyses in the prospective context in order to facilitate the evaluation of time extensions. Therefore, a majority of contractually specified methods, often called MIP 3.6 (aka "TIA"), consists of the insertion of delay events into the most current schedule update that existed at the time of the occurrence of the event: a prospective method.

At the end of the project the choices of analysis methods are expanded with the full advantage of hindsight offered by the various forms of as-built documentation. In addition, if as-built documentation is available, the best evidence rule demands that all factual investigations use the as-built as the primary source of analysis.

Also the timing distinction is often mirrored by a change in personnel. That is, often the forensic schedule analyst who typically works in the retrospective context is not the same person as the project scheduler who worked under the prospective context.

B. Layer 2: Basic Methods

The second hierarchy layer is the basic method consisting of two branches: observational and modeled. The distinction drawn here is whether the analyst's expertise is utilized for the purpose of interpretation and evaluation of the existing scheduling data only, or for constructing simulations and the subsequent interpretation and evaluation of the different scenarios created by the simulations. The distinction between the two basic methods becomes less defined in cases where the identity of the forensic analyst and the project scheduler rest in the same person.

1. Observational

The observational method consists of analyzing the schedule by examining a schedule, by itself or in comparison with another, without the analyst making any changes to the schedule to simulate any specific scenario.

Contemporaneous period analysis and as-built vs. as-planned are common examples that fall under the observational basic method.

2. Modeled

Unlike the observational method, the modeled method calls for intervention by the analyst beyond mere observation. In preparing a modeled analysis the analyst inserts or extracts activities representing delay events into or from a CPM network and compares the calculated results of the 'before' and 'after' states.

Common examples of the modeled method are the collapsed as-built, time impact analysis, and the impacted as-planned.

C. Layer 3: Specific Methods

At the third layer are the specific methods.

1. Observational Methods

Under the observational method, further distinction is drawn on whether the evaluation considers just the original schedule logic or the additional sets of progressive schedule logic that were developed during the execution of the project, often called the dynamic logic.

a. Static Logic Observation

A specific subset of the observational method, the static logic variation compares a plan consisting of one set of network logic to the as-built state of the same network.

The term 'static' refers to the fact that observation consists of the comparison of an as-built schedule to just one set of as-planned network logic.

The as-planned vs. as-built is an example of this specific method.

b. Dynamic Logic Observation

In contrast with the static logic variation, the dynamic logic variation typically involves the use of schedule updates whose network logic may differ to varying degrees from the baseline and from each other. This variation considers the changes in logic that were incorporated during the project.

The contemporaneous period analysis is an example of this specific method. Note that this category does not occur under the prospective timing because the use of past updates indicates that the analysis is performed using retrospective timing.

2. Modeled Methods

The two distinctions under the modeled method are whether the delays are added to a base schedule or subtracted from a simulated as-built.

a. Additive Modeling

The additive modeling method consists of comparing a schedule with another schedule that the analyst has created by adding schedule elements (i.e. delays) to the first schedule for the purpose of modeling a certain scenario.

The Additive Modeling methods include the impacted as-planned and some forms of the window analysis method. The MIP 3.6 (aka "TIA") can also be classified as an additive modeling method. This term or its equivalent, time impact evaluation (TIE), has been used in contracts and specifications to refer to other basic and specific methods as well.

b. Subtractive Modeling

The subtractive modeling method consists of comparing a CPM schedule with another schedule that the analyst has created by subtracting schedule elements (i.e. delays) from the first schedule for the purpose of modeling a certain scenario.

The collapsed as-built is one example that is classified under the subtractive modeling method.

D. Layer 4: Basic Implementation

The fourth layer consists of the differences in implementing the methods outlined above. The static logic method can be implemented in a gross mode or periodic mode. The dynamic logic method can be implemented as contemporaneous: as-is, contemporaneous split, contemporaneous modified, or recreated. The additive or subtractive modeling method can be implemented as a single base with simulation or a multiple base with simulation.

1. Gross Mode or Periodic Mode

The first of the two basic implementations of the static logic variations of the observational method is the gross mode. Implementation of the gross mode considers the entire project duration as one whole analysis period without any segmentation.

The alternate to the gross mode is the periodic mode. Implementation of the periodic mode breaks the project duration into two or more segments for specific analysis focusing on each segment. Because this is an implementation of the static logic method, the segmented analysis periods are ***not*** associated with any changes in logic that may have occurred contemporaneously with these project periods.

2. Contemporaneous / As-Is or Contemporaneous / Split

This basic implementation pair occurs under the dynamic logic variation of the observation method. Both choices contemplate the use of the schedule updates that were prepared contemporaneously during the project. However the as-is implementation evaluates the differences between each successive update in its unaltered state, while the split implementation bifurcates each update into the pure progress and the non-progress revisions such as logic changes.

The purpose of the bifurcation is to isolate the schedule slippage (or recovery) caused solely by work progress based on existing logic during the update period from that caused by non-progress revisions newly inserted (but not necessarily implemented) in the schedule update.

3. Modified or Recreated

This pair, also occurring under the dynamic logic variation of the observational method, involves the observation of updates. Unlike the contemporaneous pair, however, this implementation involves extensive modification of the contemporaneous updates, as in the modified implementation, or the recreation of entire updates where no contemporaneous updates exist, as in the recreated implementation.

4. Single Base, Simulation or Multi-Base, Simulation

This basic implementation pair occurs under the additive and the subtractive modeling methods. The distinction is whether when the modeling (either additive or subtractive) is performed, the delay activities are added to or extracted from a single CPM network or multiple CPM networks.

For example, a modeled analysis that adds delays to a single baseline CPM schedule is a single base implementation of the additive method, whereas one where delays are extracted from several as-built simulations is a multi-base simulation implementation of the subtractive method.

A single base additive modeling method is typically called the impacted as-planned. Similarly the single simulation subtractive method is called the collapsed as-built. The multi-base, additive simulation variation is often called a window analysis.

The nine Method Implementation Protocols (MIP) in Section 3 represent the instances of basic protocols based on the distinctions outlined in Layer 4.

E. Layer 5: Specific Implementation

1. Fixed Periods vs. Variable Periods / Grouped Periods

These specific implementations are the two possible choices for segmentation under all basic implementations except gross mode and the single base / simulation basic implementations. They are not available under the gross mode because the absence of segmentation is the distinguishing feature of the basic gross mode. They are not available under the single base / simulation basic implementation because segmentation assumes a change in network logic for each segment; the single base simulation uses only one set of network logic for the model.

In the fixed period specific implementation, the periods are fixed in date and duration by the data dates used for the contemporaneous schedule updates, usually in regular periods such as monthly. Each update period is analyzed. The act of grouping the segments for summarization after each segment is analyzed is called blocking.

The variable period/grouped period specific implementation establishes analysis periods other than the update periods established during the project by the submission of regular schedule updates. The grouped period implementation groups together the pre-established update periods while the variable period implementation establishes new periods whose lines of demarcation may not coincide with the data dates used in the pre-established periods and/or which can be determined by changes in the critical path or by the issuance of revised or recovery baseline schedules. This implementation is one of the primary distinguishing features of the variable period analysis method.

2. Global (Insertion or Extraction) vs. Stepped (Insertion or Extraction)

This specific implementation pair occurs under the single base, simulation basic implementation, which in turn occurs under the additive modeling and the subtractive modeling specific methods. Under the global implementation delays are either inserted or extracted all at once, while under the stepped implementation, the insertion or the extraction is performed sequentially (individually or grouped).

Although there are further variations in the sequence of stepping the insertions or extractions, usually the insertion sequence is from the start of the project towards the end, whereas stepped extraction starts at the end and proceeds towards the start of the project.

1.5. Underlying Fundamentals and General Principles

A. Underlying Fundamentals

At any given point in time on projects, certain work must be completed at that point in time so the completion of the project does not slip later in time. The industry calls this work, “critical work.” Project circumstances that delay critical work will extend the project duration. Critical delays are discrete, happen chronologically, and accumulate to the overall project delay at project completion.

When the project is scheduled using CPM scheduling, the schedule typically identifies the critical work as the work that is on the “longest” or “critical path” of the schedule’s network of work activities. The performance of non-critical work can be delayed for a certain amount of time without affecting the timing of project completion. The amount of time that the non-critical work can be delayed is "float" or "slack" time referring to as Total Float

A CPM schedule for a particular project generally represents only one of the possible ways to construct the project. Therefore, in practice, the schedule analyst must also consider the assumptions (work durations, logic, sequencing, and labor availability) that form the basis of the schedule when performing a forensic schedule analysis. This is particularly true when the schedule contains preferential logic (i.e., sequencing which is not based on physical or safety considerations) and resource assumptions. This is because both can have a significant impact on the schedule's calculation of the critical path and float values of non-critical work at a given point in time.

CPM scheduling facilitates the identification of work as either critical or non-critical. Thus, at least in theory, CPM schedules give the schedule analyst the ability to determine if a project circumstance delays the project or if it just consumes float in the schedule assuming that float is not specifically owned by either party under that terms of the contract. For this reason, delay evaluations utilizing CPM scheduling techniques are now preferred for the identification and quantification of project delays.

The critical path and float values of uncompleted work activities in CPM schedules change over time as a function of the progress (or lack of progress) on the critical and non-critical work paths in the schedule network. Only project circumstances that delay work that is critical when the circumstances occur extend the overall project. Thus, when quantifying actual project delay, the accuracy in quantification is increased when the impacts of potential causes of delay are evaluated within the context of the schedule in effect at the time when the impacts happen.

B. General Principles

1. Use CPM Calculations

Calculation of the critical path and float must be based on a CPM schedule with proper logic (see Subsection 2.1.)

2. Concept of Data Date Must be Used

The CPM schedule used for the calculation must employ the concept of the data date or status date. Note that the critical path and float can be computed only for the portion of the schedule forward (future) of the data date.

3. Shared Ownership of Network Float

In the absence of contrary contractual language, network float, as opposed to project float, is a shared commodity between the owner and the contractor. In such a case float must be shared in the interest of the overall project rather than to the sole benefit of one of the parties to the contract.

4. Update Float Preferred Over Baseline Float

If reliable updates exist, relative float values for activities in those updates at the time the schedule activity was being performed are considered more reliable compared to relative float values in the baseline for those same activities.

5. Sub-Network Float Values

What is critical in a network model may not be critical when a part of that network is evaluated on its own, and vice versa. The practical implication of this rule is that what is considered critical to a subcontractor in performing its own scope of work may not be critical in the master project network. Similarly, a schedule activity on the critical path of the general contractor's master schedule may carry float on a subcontractor's sub-network when considered on its own.

6. Delay Must Affect the Critical Path

In order for a claimant to be entitled to an extension of contract time for a delay event (and further to be considered compensable), the delay must affect the critical path. This is because before a party is entitled to time-related compensation for damages it must show that it was actually damaged. Because conventionally a contractor's delay damages are a function of the overall duration of the project, there must be an increase in the duration of the project.

7. All Available Schedules Must Be Considered

Regardless of the method selected for analysis, all available sources of planning and schedule data created during the project, including but not limited to, various versions of baselines, updates and as-builts, should be examined and considered, even if they are not directly used for the analysis.

2. SOURCE VALIDATION

The intent of the source validation protocols (SVP) is to provide guidance in the ***process*** of assuring the validity of the source input data that forms the foundation of the various forensic schedule analysis methodologies discussed in Section 3. Any analysis method, no matter how reliable and meticulously implemented, can fail if the input data is flawed. The primary purpose of the SVP is to minimize the failure of an analysis method based upon the flawed use of source data.

The approach of the SVP is to maximize the ***reliable use*** of the source data as opposed to assuring the underlying reliability or accuracy of the substantive content of the source data. The best accuracy that an analyst can hope to achieve is in the faithful reflection of the facts as represented in contemporaneous project documents, data, and witness statements. Whether that reflection is an accurate model of reality is almost always a matter of debatable opinion.

Source validation protocols consist of the following:

2.1. Baseline Schedule Selection, Validation, and Rectification (SVP 2.1)
2.2. As-Built Schedule Sources, Reconstruction, and Validation (SVP 2.2)
2.3. Schedule Updates: Validation, Rectification, and Reconstruction (SVP 2.3)
2.4. Identification and Quantification of Discrete Impact Events and Issues (SVP 2.4)

2.1. Baseline Schedule Selection, Validation, and Rectification (SVP 2.1)

A. General Considerations

The baseline schedule is the starting point of most types of forensic schedule analyses. Even methods that do not directly use the baseline schedule, such as the modeled subtractive method, often refer to the baseline for activity durations and initial schedule logic. Hence, assuring the validity of the baseline schedule is one of the most important steps in the analysis process.

Note that validation for forensic purposes may be fundamentally different from validation for purposes of project controls. What may be adequate for project controls may not be adequate for forensic scheduling, and vice versa. Thus, the initial focus here is in assuring the functional utility of the baseline data as opposed to assuring the reasonableness of the information that is represented by the data or optimization of the schedule logic. So, for example, the validation of activity durations against quantity estimates is probably not something that would be performed as part of this protocol. The test is that if it is possible to build the project in the manner indicated in the schedule and still be in compliance with the contract, then do not make any subjective changes to improve it or make it more reasonable.

The obvious exception to the above would be where the explicit purpose of the investigation is to evaluate the reasonableness of the baseline schedule for planning, scheduling and project control purposes. For those guidelines please refer to other Recommended Practices published by AACE[3].

The recommended protocol outlined below assumes that the forensic analysis contemplates the investigation of schedule deviations at Level 3 (sufficient detail to monitor and manage the overall project) degree of detail[4]. The user is cautioned that an investigation of schedule

[3] AACE International's Planning & Scheduling Committee is developing an RP that includes an extensive discussion on the baseline schedule.
4 Refer to AACE International's RP 37R-06 for additional information on Schedule Levels of Detail.

deviations at Level 1 or 2 may require less detail. Similarly, investigation of schedule deviations at Level 4 may require verification at a higher level of detail.

The recommended protocol below is worded as a set of investigative issues that should be addressed. If the baseline schedule is to be used in an observational analysis, the forensic schedule analyst may simply note the baseline's schedule's compliance or non-compliance with the various protocols below. If however, the baseline schedule is to be used in a modeled analysis, the various protocols below form the basis for documented alterations so that the adjusted baseline schedule both reflects its original intent as closely as possible and still meets the procedural elements of the recommended protocol.

SVP 2.1 also forms the basis of SVP 2.3, which deals with the validation and rectification of schedule updates, since early updates are based almost entirely on the baseline schedule.

B. Recommended Protocol

CAVEAT: When implementing MIPs 3.3 or 3.4, baseline validation protocols involving changes to logic or calendars should ***not*** be implemented.

1. Ensure that the work breakdown and the level of detail are sufficient for the intended analysis.

2. Ensure that the data date is set at notice-to-proceed (or earlier) with no progress data for any schedule activity that occurred after the data date.

3. Ensure that there is at least one continuous critical path, using the longest path criterion that starts at the earliest occurring schedule activity in the network (start milestone) and ends at the latest occurring schedule activity in the network (finish milestone).

4. Ensure that all activities have at least one predecessor, except for the start milestone, and one successor, except for the finish milestone.

5. Ensure that the full scope of the project/contract is represented in the schedule.

6. Investigate and document the basis of any milestones dates that violate the contract provisions.

7. Investigate and document the basis of any other aspect of the schedule that violates the contract provisions.

8. Document and provide the basis for each change made to the baseline for purposes of rectification.

9. Ensure that the calendars used for schedule calculations reflect actual working day constraints and restrictions actually existing at the time when the baseline schedule was prepared.

10. Document and explain the software settings used for the baseline schedule.

C. Recommended Enhanced Protocol

CAVEAT: When implementing MIP 3.3 or 3.4, baseline validation protocols involving changes to logic or calendars should ***not*** be implemented.

1. The level of detail is such that no single schedule activity carries a contract payment value of more than one half of one percent (½%) of total contract payment value per unit of activity duration, and no more than five percent (5%) of total contract payment value per schedule activity.

2. Create separate activities for each responsible party.

3. Document the basis of all controlling and non-controlling constraints.

4. Replace controlling constraints, except for the start milestone and the finish milestone, with logic and/or activities.

5. Because delay scenarios often involve factors external to the original contract assumptions when the baseline was created, it may be necessary to add activities or enhance the level of detail beyond that contained in the baseline.

6. If the description of the schedule activity is too general or vague to properly ascertain the scope, the schedule activity should be subdivided into detailed components using other progress records.

D. Special Procedures

1. Summarization of Schedule Activities

a. If the level of detail of the baseline is clearly excessive in comparison to the delays being evaluated, the analyst may choose to summarize the baseline schedule for purposes of analysis. In doing so, the following guidelines are recommended:

b. Ensure that summarization is restricted to activities that do not fall on the critical or near-critical paths.

c. Organize the full-detail source schedule so that the identity of the activities comprising the summary schedule activity can be determined using:

   i. Summarizing or hammocking.

   ii. Work breakdown structure (WBS).

   iii. Coding of the detail activities with the summarized activity ID.

d. Restrict the summarization to logical chains of activities with no significant predecessor or successor logic ties to activities outside of the summarized detail.

e. Restrict the summarization to logical chains of activities that are not directly subject to delay impact evaluation or modeling.

2. Reconstruction of a Computerized CPM Model from a Hardcopy

a. The recommended set of hardcopy data necessary for an accurate reconstruction is:

   i. Predecessor & successor listing with logic type and lag duration, preferably sorted by activity ID.

ii. Tabular listing of activities showing duration, calendar ID, early and late dates, preferably sorted by activity ID.

iii. Detailed listing of working days for each calendar used.

b. The recommended level of reconstruction has been reached when the reconstructed model and the hardcopy show matching data for:

i. Early start & early finish.

ii. Late start & late finish.

c. A graphic logic diagram alone is not a reliable hardcopy source to reconstruct an accurate copy of a schedule.

3. De-statusing a Progressed Schedule to Create a Baseline

If a baseline schedule is not available, but a subsequent CPM update exists, the progress data from the update can be removed to create a baseline schedule. Also, the schedule that is considered to be the baseline schedule may contain some progress data or even delays that occurred prior to the preparation or the acceptance of the baseline schedule. The general procedure consists of the following:

a. For each schedule activity with any indicated progress, remove actual start (AS) and actual finish (AF) dates.

b. For each schedule activity with any indicated progress, set completion percentage to 0%.

c. For each schedule activity with any indicated progress, set remaining duration (RD) equal to original duration (OD).

i. The OD should be based on the duration that was thought to be reasonable at the time of NTP. If the update is one that was prepared relatively early in the project, it is likely that the OD is the same as the OD used in the baseline schedule.

ii. The OD should not be based on the actual duration of the schedule activity from successive updates.

d. Set the schedule data date (DD) to the start of the project, usually the notice-to-proceed or some other contractually recognized start date.

e. Review the scope of the progressed schedule to determine whether it contains additions to or deletions from the base contract scope. If so, modify the schedule so it reflects the base contract scope.

4. Software Format Conversions

a. Document the exact name, version, and release number of the software used for the source data which is to be converted.

b. If available, use a built-in automatic conversion utility for the initial conversion and compare the recalculated results to the source data for:

i. Early start & early finish.

ii. Late start & late finish.

c. Manually adjust for an exact match of the early and late dates by adjusting:

i. The lag value of a controlling predecessor tie and the calendar assigned to the lag value, if necessary.

ii. The relationship type of a controlling predecessor tie.

iii. Activity duration.

iv. Constraint type and/or date.

d. Document all manual adjustments made and explain and justify if those adjustments have a significant effect on the network.

2.2. As-Built Schedule Sources, Reconstruction, and Validation (SVP 2.2)

A. General Considerations

Along with the baseline schedule, the as-built schedule, more specifically the as-built schedule data, is one of the most important source data for most types of forensic schedule analysis methods. Even methods that do not directly use the as-built schedule, such as the modeled additive methods, often refer to the as-built schedule data to test the reasonableness of the model. As with the baseline, assuring the validity of the as-built schedule data is one of the most important steps in the analysis process.

It is important to accept the fact that the accuracy and the reliability of as-built data are never going to be perfect. Rather than insisting on increasing the accuracy, it is better to recognize uncertainty and systematize the measurement of the level of uncertainty of the as-built data and document the source data. One of the simplest systems is to call all uncertainty in favor of the adverse party. However, it may be more defensible to use a set of consistent set of documentation for the as-built. Of course the most reasonable solution may be for both parties to agree on a set of as-built dates prior to proceeding with the analysis and the resolution of the dispute.

There are two different approaches to creating an as-built schedule. The first one is to create an as-built schedule from scratch using various types of progress records, for example, the daily log. The resulting schedule is defined by and potentially constrained by the level of detail and the scope of information available in the project records used to reconstruct the as-built.

The second approach is to adopt the fully progressed update as the basic as-built schedule and modify or augment it as needed. Often a fully progressed update is not available and the analyst must complete the statusing of the schedule using progress records. A subset of this approach is to create a fully progressed baseline schedule from progress records. In implementing this approach it is important to understand the exact scope of the activities in the baseline schedule before verifying or researching the actual start and finish dates.

The subtractive modeling methods require an as-built schedule with complete logic as the starting point. Note that the preparation of the model requires not only the validation of as-built dates but also the simulation of an as-built schedule based on actual durations, logic and lags.

To qualify as an as-built schedule, the cause of delays need not be explicitly shown so long as the delay effect is shown. For example, if a scheduled activity that was planned to be completed in ten days but took thirty days and is shown as such, the cause of the delay need not be shown for it to be a proper as-built. However, as the analysis progresses, eventually the delay causation would need to be addressed and made explicit in some form. Note that if the analyst chooses to explicitly show delays, SVP 2.4 covers the subject of identification and quantification of delays.

In most cases the as-built schedule is a fully statused scheduled with a data date equal to or later than the actual completion date of the project. However, the term “as-built” may also be used to describe the most recent schedule update. In this case, only the activities which are statused to the left of the data date are considered “as-built” data. Consequently it is possible to perform a comparative as-built analysis, such as MIP 3.1, prior to the actual completion of the overall project, as long as the delaying events and its effects have all occurred prior to the data date.

The as-built critical path cannot be directly determined using conventional float calculation on the past portion (left) of the data date. Because of this technical reason, often the critical set of as-built activities is called the ***controlling*** activities as opposed to ***critical*** activities.

Objective identification of the controlling activities is difficult, if not impossible, without the benefit of any schedule updates or at least a baseline CPM schedule with logic. Therefore, in the absence of competent schedule updates, the analyst must err on the side of over-inclusion in selecting the controlling set of as-built activities. The determination must be a composite process based on multiple sources of project data including the subjective opinion of the percipient witnesses.

Contemporaneous perception of criticality by the project participants is just as important as the actual fact of criticality because important project execution decisions are often made based on perceptions. For more on the subject of Identifying the As-Built Critical Path, refer to Subsection 4.3.C.

The recommended protocol outline below assumes that the forensic analysis contemplates the investigation of schedule deviations at Level 3 (project controls) degree of detail. The user is cautioned that an investigation of schedule deviations at Level 1 or 2 may require less detail. Similarly, an investigation of schedule deviations at Level 4 may require verification at a higher level of detail.

B. Recommended Protocol

1. If a schedule update is the primary source of as-built schedule data:

    a. Ensure that the data date is set equal to or later than the events and impacts that are to be evaluated in the analysis.

    b. Ensure that all activities to the left of the data date have actual start and completion dates.

c. Ensure that all activities to the right of the data date do not have actual start or finish dates.

d. Perform a check of the as-built dates using the source deemed most reliable other than the update itself.

e. If possible, interview the project scheduler or other persons most knowledgeable for updated data collection and data entry procedures to evaluate the reliability of the statusing data.

f. Determine and allow for whether significant changes have been made to activity descriptions and IDs.

g. Understand the exact scope and assumptions underlying the schedule activities so that the as-built data is a reflection of the same scope and assumptions.

2. Perform a check of all critical and near-critical activities as defined by this RP and a random 10% sampling of all activities against the reliable alternate source to determine whether a more extensive check is necessary. Note that this step may have to be repeated as ongoing analysis warrants the inclusion of more activities as critical or near-critical than originally identified.

3. Dates of <u>significant</u> activities should be accurate to 1 working day and dates of all other activities should be accurate to 5 working days or less.

4. Contractual dates such as notice-to-proceed, milestones, and completion dates should be accurate to the exact date. Should those dates be subject to dispute, the justification for the selection of the dates should be clearly stated.

C. Recommended Enhanced Protocol

1. Tabulate all sources of as-built schedule data and evaluate each for reliability.

2. If a baseline schedule exists and where a direct comparison between the baseline and the as-built would be difficult due to changes in activity IDs, descriptions, and/or software packages, an "as-built" can be created by fully progressing the planned activities allowing for a one-to-one planned versus actual comparison of each baseline schedule activity. See Subsection 2.2.D.2.

3. Show discrete activities for delay events and delaying influences.

4. If the description of the schedule activity is too general or vague to properly ascertain the scope, the schedule activity should be subdivided into detailed components using other progress records.

D. Special Procedures

1. Creating an Independent As-Built from Scratch "Daily Specific As-Built" (DSAB)

a. An as-built record of the work on a project is often necessary to verify the accuracy of the CPM dates reflected in the various schedule updates and to identify and correlate events inside a single CPM schedule activity. This identification of events inside a CPM schedule activity is essential to particularize possible shifts in the schedule and explain responsibility for any delays.

b. The best source for as-built data is a continuous daily history of events on the project developed and maintained by persons working on the project. Traditionally, there are contractor's daily reports, but there may also be owner's daily inspection reports or a scheduler's daily progress report. These daily records can be augmented as required by other primary sources such as certified payrolls and timesheets, completion certificates, inspection reports, incident reports, and start-up reports. Secondary sources such as weekly meeting minutes or progress reports can also provide insight into what happened.

c. It is often best to develop the DSAB using a database where every entry in the daily report is separately listed as a record. Such a database would allow for the complete history of each schedule activity over time, or an electronic version of the daily report coded for activities worked on each particular day. Notes on the daily reports such as problems or delays can be listed as additional activities.

d. It is important to develop a correlation between as-planned activities and as-built activities. Baseline schedule activities usually include descriptions sufficient to distinguish them from other similar activities. The as-built schedule is coded to the same activities included in the baseline schedule. It is frequently the case that there is not a perfect match between the activities of the two schedules. Some of the as-planned activities do not appear in the as-built, and, more frequently, there are significant as-built activities that are either in greater detail than the as-planned or simply do not appear in the as-planned.

   i. *Activity in the baseline schedule, but not the as-built schedule*--There are generally three reasons for an activity to appear in the baseline schedule but not the as-built schedule. The first and most likely reason is that the as-built is not sufficiently detailed. This is usually because the work depicted in detail in the baseline schedule is described more generically in the as-built. In this case, the preferred method would be to divide the as-built activity into two constituent parts if contemporaneous notes permit. If this is not possible, then the two represented activities in the baseline schedule should be combined. The second reason could be that the schedule activity was deleted by change order and thus does not appear in the as-built. If this is the case, it is generally not appropriate to modify the baseline schedule. Rather, the lack of an as-built activity will have to be evaluated in light of successor work. The third reason rarely occurs: The contractor may not have performed a specific aspect of the work, even though it is required. In such a situation the longer duration of the predecessor or successor must be considered in light of the "missing" schedule activity.

   ii. *Activities in the as-built schedule, but not the baseline schedule --There generally are* three reasons for an activity appearing in the as-built schedule but not the baseline schedule. The first and most likely possibility is that the actual activity is simply reported in more detail in the as-built than in the as-planned. In this situation, it is generally better to combine the more detailed as-built data into a schedule activity that is reflected in the as-planned. However, this extra detail from the as-built may be necessary in performing a responsibility analysis. The second reason could be that the activity was new--it was added by a change order. If this is the case, it is generally again not appropriate to modify the baseline schedule. Rather, the new as-built activity should be treated simply as additional work and coded in such a manner as to indicate this situation and permit the analysis to properly consider it. The third reason is that the baseline schedule might not completely reflect the actual scope of contractual work.

Again, it is probably best not to alter the baseline schedule but rather to reflect the actual work activity in its proper logical as-built sequence. This should not occur if the analysis is utilizing a properly validated baseline schedule (see SVP 2.1).

e. Line up the as-built and baseline schedule--This step can be performed either in a large database with graphical output, or can be done in a more personal/mechanical manner by hand.

i. *Using a database*--By using a database, the analyst can arrange or cluster the activities according to whatever sequence seems most appropriate. For example, it may be useful on a multi-building project to review the data by building. Alternatively, if the performance of a particular trade is important, then the review could be performed based on trade. It is possible through export from a database to a graphical program to plot the baseline schedule data (early/late, start/finish) directly against the as-built record.

ii. *By hand. (A.K.A. X-chart or Dot-chart)*--On small projects it is possible to simply plot the data graphically by hand. This technique is called the "X-chart" because the analyst placed an "X" in the appropriate date and activity of a chart with dates along the X-axis and activities along the Y-axis. This pre-computer technique is still useful for smaller projects or partial analysis.

f. Identify the true "start" of an activity--It is usually relatively easy to identify from the as-built data the start of an activity but not always. It is recommended that the start of an activity be considered the first date associated with a series of substantive work days on the activity. Care should be taken in discounting "false starts" or "false finishes" that they do not reflect a true delay. Care should also be taken to ensure that a false start does not actually represent an actual start coupled with a suspension due to a delay event.

g. Identify the true "finish" of an activity--The same logic as above applies to the finish dates. Generally the analyst, absent specific data to the contrary, should assume that when the period of concentrated work is completed on an activity, the activity is complete. Another possible criterion is that an activity can be considered *logically* complete when a successor tied with a simple FS logic is able to start substantive work.

2. Creating a Fully Progressed Baseline

a. A fully-progressed version of the baseline schedule allows for a comparison of the plan to actual performance at an individual activity level of detail. Often, however, a progressed baseline is not readily available because the schedule is changed during progress.

b. The most expedient procedure to create a fully progressed baseline is to use the as-built data for each activity contained in the final update and transfer them to the corresponding baseline activities. In implementing this procedure the analyst must:

a. Recognize that using the activity ID as the sole criterion for correspondence between the final update and the baseline may not be adequate if the activity descriptions are not virtually identical.

b. Therefore, in addition, the analyst must understand the scope and the assumptions underlying the baseline schedule activities so that the as-built data is a reflection of the same scope and assumptions.

c. The baseline set of activities may have to be summarized to receive the corresponding as-built data if the activities have been summarized in the final update.

d. If the corresponding final update activities are more detailed than the baseline activities, determine the update activity representing the start of the less detailed activity chain in the baseline and the update activity representing the finish of that same chain in order to set the actual start and finish dates.

3. Determination of 'Significant' Activities for Inclusion in an As-Built

Many CPM schedules in current use contain hundreds if not thousands of activities. While that level of detail may be necessary to keep track of performance and progress for the purpose of project controls, the facts of the dispute may not require the analysis of each and every activity in a forensic context. This section offers guidelines for streamlining and economizing the as-built analysis process without compromising the quality of the process and the reliability of the results.

Because this step typically occurs early in the analysis process, the analyst may not have a full understanding of the project and the issues. Therefore, the criterion is of ***prima facie*** significance. In other words, if in doubt, consider it significant. As a result, it is possible that at the end of the analysis some of the selected activities are considered insignificant. But that is better than discovering at the end of the analysis that some significant activities and key factors were not considered. This is a multi-iterative process that requires continuous refinement of the set of significant activities during the analysis process.

The main factor for significance is criticality. The procedure for determining the as-built critical path is discussed in Subsection 4.3.C and the procedure for determining the significant activities includes the procedure set forth in Subsection 4.3.C. However, <u>in addition</u> to those items the following items are recommended for inclusion in the significant set:

- Suspected concurrent delays including those alleged by the opposing party

- Activity paths for which time extensions were granted

- Delay events and all activities on the logical path(s) on which those events lie

- All milestones used in the schedule

- High-value (based on pay loading) activities

- High-effort (based on resource loading) activities

Note that in many cases some significant activities are not discretely and explicitly contained in the CPM model. Obviously, these extraneous activities must also be considered in the as-built.

4. Collapsible As-Built CPM Schedule

The fundamental difference between a fully progressed CPM and a collapsible as-built CPM schedule is in the schedule logic. The fully progressed CPM schedule can graphically illustrate the as-built condition using the actual start and actual finish dates assigned to each schedule activity. However, the schedule cannot be used for calculation because it has been fully progressed. Therefore, the actual activity duration (AD) and the logic ties are no longer controlling the network calculation. On the other hand, the collapsible as-built is a CPM model of the as-built condition. The schedule logic is revised by assigning actual durations to the activities and tying them together with logical relationships so that the actual start and the actual finish dates are simulated in the schedule as calculated start and finish dates. For a step-by-step procedure please refer to MIP 3.8.

5. Summarization of Schedule Activities

a. If the level of detail of the as-built is clearly excessive in comparison to the delays being evaluated, the analyst may choose to summarize the as-built schedule for purposes of analysis. In doing so, the following guidelines are recommended:

b. Ensure that summarization is restricted to activities that do not fall on the critical or near-critical paths.

c. Organize the full-detail source schedule so that the identity of the activities comprising the summary schedule activity can be determined using:

   i. Summarizing or hammocking.

   ii. Work breakdown structure (WBS).

   iii. Coding the detail activities with the summarized activity ID.

d. Restrict the summarization to logical chains of activities with no significant predecessor or successor logic ties to activities outside of the summarized detail.

e. Restrict the summarization to logical chains of activities that are not directly subject to delay impact evaluation or modeling.

2.3. Schedule Updates: Validation, Rectification, and Reconstruction (SVP 2.3)

A. General Considerations

SVP 2.3 discusses issues involved in evaluating the project schedule updates for use in forensic schedule analysis.

The schedule update consists of the as-built portion on the left side of the data date, the as-planned portion on the right side of the data date, and the data date itself. Because SVP 2.1 addresses the issues relevant to the as-planned portion, and 2.2 addresses the issues relevant to the as-built portion, the focus of SVP 2.3 is on the practice of updating the schedule with progress information and the reliable use of that progress data.

B. Recommended Protocol

1. Interview the project scheduler or other persons-most-knowledgeable for updated data collection and data entry procedures to evaluate the reliability of the statusing data.

2. Assemble all schedule updates so that they cover the entire project duration from start to finish or up to the current real-time data date.

3. Use officially submitted schedule updates.

4. Ensure that the update chain starts with a recognized baseline.

5. Check on the consistency of the actual start and finish dates assigned to each schedule activity from update to update.

6. Document and provide the basis for each update, noting all changes made that extend, reduce, or change the longest path or the controlling path to an interim contractual milestone.

7. If other progress records are available, check the remaining duration and percentage complete values for accuracy and reasonableness.

C. Recommended Enhanced Protocol

1. Implement SVP 2.1 for the as-planned portion of each schedule update, including the baseline.

2. Implement Subsection 2.4.D.2. to bifurcate the pure-progress step from the logic revision steps in each update.

D. Special Procedures

1. Reconstructed Updates

There are two main schools of thought on recreating a partially statused schedule. The first school of thought, called the hindsight method, states that since the forensic scheduler is performing the analysis after the job has been completed, the analyst should use the actual performance dates and durations to recreate the updates.

The second school of thought, called the blinders or the blindsight method, requires the analyst to pretend that the analyst does not have access to actual performance data and simulate the project scheduler's mindset at the time the update was actually being prepared. Therefore, the analyst needs to consider what the scheduler would have assigned as the remaining duration for that schedule activity at that time. If the analyst cannot logically make that guess, the analyst needs to be as objective as possible and follow a remaining duration formula.

Outlined below are the two methods:

a. "Hindsight Method"

In this method, the actual status of the schedule activity in the succeeding schedule update period is used to calculate the remaining duration of the previous schedule update. This is delineated in the formula below:

i. RD = actual duration of succeeding update - (data date - actual start of activity) where the data date is the data date of the existing schedule update that needs to be statused.

b. "Blindsight" Method

In this method, it is assumed that the analyst does not have the update schedule for the succeeding period and has no knowledge of the project conditions later than the update under investigation. Therefore, the analyst must stand in the shoes of the scheduler at the time of the project. Note that the progress curve created by this method assumes a straight line.

i. IF: data date (DD) - actual start of the activity (AS) < original duration (OD), THEN: remaining duration (RD) = OD - (DD - AS)

ii. IF: DD - AS > OD, THEN: RD = 1

2. Bifurcation: Creating a Progress-Only Half-Step Update

Bifurcation (aka half-stepping or two-stepping) is a procedure to segregate progress reporting from various non-progress revisions inherent in the updating process. This should not be considered a revision or modification of the update schedules but rather a procedure that examines selected data, namely logic changes isolated by this process, which may be present in the updates of record. For a step-by-step implementation of the bifurcation process refer to MIP 3.4

3. Changing the Contemporaneous Project Schedule For the Analysis

Due to the complex nature of construction projects and the fact that CPM schedules are models of reality, not reality itself, the analyst will inevitably encounter an instance when the contemporaneous project schedule contains an error that could affect the assessment of critical project delay. Instead of completing the analysis using a schedule with an error or entirely abandoning the schedules because of the error, the analyst has the option to correct the error in the contemporaneous project schedule and use the corrected schedule as the basis for the analysis.

Correcting the contemporaneous schedules does not automatically result in a shift in classification of the analytical technique from an analysis based on contemporaneous schedules such as MIP 3.3 (Observational / Dynamic / As Is) to one based on non-contemporaneous schedules such as MIP 3.5 (Observational / Dynamic / Modified or Recreated).

Having stated that, the preference of every analyst should be to use the contemporaneous schedules and updates as they were prepared, reviewed, approved or accepted, and used on the project. This belief is grounded in the fact that the parties used the imperfect schedules to make decisions and manage the project work. Thus, these schedules, even though not perfect, are the best representation of the parties' objectives and understanding of the project contemporaneously and are an indicator of each party's performance.

However, absent contract language mandating the use of the contemporaneous schedules to quantify delay, corrections to the contemporaneous schedules can be properly considered by the analyst without eroding the credibility of the resulting analysis. The following is a discussion of examples of revisions to the contemporaneous schedules that may fit within the boundaries of such corrections:

a. Correcting a Wrong Actual Start or Finish Date

Sometimes, the actual start and finish dates recorded in the contemporaneous project schedules may be inaccurate. The analyst may consider relying on other contemporaneous documents to correct these dates. The analyst may limit the correction of the wrong actual start and finish dates to paths of work that have the potential to delay the project and are on critical or near-critical paths. When an analyst chooses to correct a wrong actual date in the schedule, the analyst should be mindful that correcting a date may result in a shift in the critical path and that just because a date is wrong doesn't mean that it must be corrected. If the project team never recognized that the date was wrong, then the team relied on the schedule generated by calculations based on that date, and the correction should not be made because it would not then represent the mindset of the team on which decisions were made at the time.

b. Correcting Minor Schedule Errors

In deciding whether it is appropriate to make corrections for minor schedule errors, the analyst should verify that the project participants recognized the error in the schedule contemporaneously and that the error was ultimately corrected by the project team in the contemporaneous project schedules at some point during the project. If the error was not corrected in contemporaneous project schedules, the error should have at least been recognized and identified as an error in the schedule in a contemporaneous project document. A minor schedule error may include:

i. An incorrect logic relationship
ii. A missing logic relationship
iii. An incorrect activity based on described scope of the activity
iv. A missing activity

c. Bringing a Revision Back in Time to Represent Added or Changed Work

This situation occurs when a schedule revision or fragnet (fragmentary network representing added or changed work) was inserted into the contemporaneous project schedules well after the change or event that necessitated the revision occurred. If the schedule revision or fragnet was not inserted into the appropriate contemporaneous project schedule, but was recognized and identified in a contemporaneous project document as a change that should have been made, then the analyst may decide to insert the schedule revision or fragnet into the contemporaneous schedule update in effect when the change occurred to measure the resulting delay.

This correction involves bringing back (or inserting) the schedule revision or fragnet to the point or nearest the moment in time) when the event occurred. The schedule revision or fragnet that is brought back in time (or inserted) will typically be a duplicate of an existing revision or fragnet that was inserted into the schedule during the project or as described in the contemporaneous project documents. It must be noted that if the fragnet consists of actual durations, and relationships, this procedure would create a hindsight impact simulation as opposed to a blindsight impact simulation, which would be implemented with a fragnet consisting of planned durations and relationships estimated at the time the event occurred.

d. Splitting an Activity

Typically, updates increase in detail as the schedule progresses, therefore the number of activities increase, not necessarily an increase in scope but an increase in detail. When a variance analysis is performed between two updates with different activity counts, exact correlation is not possible since the more detailed activity set did not exist in the previous update. Therefore, the detailed activity set should be replicated in the previous update with the same planning duration, logic and dates of the summary activity.

All of these corrections should be described in the analyst’s report along with the basis of the corrections so that the other parties and the fact finders understand the changes that the analyst made to the contemporaneous schedule.

When an analyst concludes that more extensive revisions are necessary to the contemporaneous project schedules than those contemplated in paragraphs a., b., and c., above, such revisions should be made cautiously, consistently, and founded to the greatest extent possible on the contemporaneous project documentation. The analyst must also remember that most schedules are models and, hence, perfection is not the standard.

The issue of correcting the schedule is one of balance and reasonableness and, for these reasons corrections should not be made across the board or automatically.

Note that some significant errors in the underlying analysis schedules may not substantially affect the ultimate conclusions of the analysis. For example, imagine a schedule where a significant activity was omitted. Even though the work is absent from the schedule, it would not necessarily be absent from the analysis. If three activities, A, B, and C, must be performed in sequence, but the schedule leaves out B, the analysis will still detect a delay due to B. This is because C cannot start until B is completed. Any delay attributable to B will show up as a delay to the start of C. There may be no need to “correct” the schedule by adding B into the schedule. Delays to B may be addressed by the analysis even though B is not present.

Finally, the analyst must also be consistent and maintain independence and objectivity. The analyst cannot limit its corrections to those that have the affect of improving the analyst’s client’s position.

2.4. Identification and Quantification of Discrete Delay Events and Issues (SVP 2.4)

A. General Considerations

SVP 2.4 discusses the compilation of information regarding delay events, activities and influences that are inserted or extracted in modeled methods or used in evaluating the observational methods. As stated in the introduction to the SVP, the approach of the SVP is to maximize the reliable use of the source data as opposed to assuring the reliability or the accuracy of the substantive content of the source data. The best accuracy that an analyst can hope to achieve is an objective reflection of the facts as represented in documents, data and witness statements. Whether that reflection is an accurate model of reality is almost always a matter of debatable opinion. This is especially true in assembling delay data and making the causal connection between the delay event or influence and the impacted activity.

1. 'Delay' Defined

For the purpose of this section, the term, 'delay', is considered neutral in terms of liability. Delay simply means a state of extended duration of an activity, or a state of an activity not having started or finished on time, relative to its predecessor.

a. Activity-Level Variance (ALV)

Forensic delay analysis primarily focuses on determining start or duration variances of a specific schedule activity otherwise known as activity-level variances or ALV's.

ALV's are the result of several types of delay causes:

- Waiting (delayed start)
- Performance (Productivity Impacts, Additional Work, etc.)
- Interruption (Work Stoppage, Weather, Strikes, etc.)

For example a delayed start of an activity awaiting a response to an RFI is the delay cause "waiting." In contrast, a delayed start due to the performance of a scope of work that was missed at bid time is the performance of additional scope of work. Finally, an activity experiencing numerous rain days over several months is experiencing interruption of work or otherwise known as disruption. Given these variations there are two main ways in which ALVs are expressed in a CPM schedule:

i. Delayed Relative Start. This is the variance between the planned start relative to the planned controlling predecessor to the actual point of start. Because this is a relative measure, it cannot be determined by the comparison of planned date (either early or late) to the actual, which would yield a cumulative delay figure. The cumulative delay incorporates all the delays that occurred previously in the activity chain.

ii. Extended Duration. An extended duration delay occurs when the actual activity duration exceeds the planned original duration or reasonable duration required to perform the described activity. Unlike the delayed relative start case, extended duration calculations are not dependent on predecessor logic for quantification. Extended durations may result from continuous impact, intermittent impact such as stop-and-go operations, weather delays, or from discrete periods of added work or suspensions. In addition, extended durations may be due to experiencing lower labor productivity than planned for when the activity duration was developed. Unless the delay is fully attributable to a discretely identifiable period of exclusive extra work performance, quantification of this type of delay requires some estimating on the part of the analyst.

b. **Distinguishing ALV from Project-Level Variance (PLV)**

The ALV should be distinguished from the project-level variance (PLV) which is also a variance but at the overall project level. While the ALVs occur close in time to the causes, i.e. in the same period, the PLV may be months apart from the actual cause(s of the delay PLV is the result of the aggregation of ALV's after taking into account network float. Within the context of this RP, ALVs are considered 'delays' regardless of the amount of float they carry. The activity experiences a delay if an ALV exists regardless if the delay affects the project completion date, i.e. the PLV.

c. Distinguishing Delay-Cause from Delay-Effect

It is important for the analyst to be able to distinguish the cause of delay from the resulting effect. For example, a fully updated schedule may show extended activities and delayed start of activities relative to their controlling predecessors. While the cause may not be apparent, a competent statusing of the schedule will show the delay-effects. What caused the initial ALV for the chain of activities often does not appear on the schedule but must be investigated and researched using project documents, data and witness interviews. If, on the other hand, a delay was appropriately inserted into the schedule as a new activity as a predecessor to the activity with the start delay, the cause of the ALV is readily apparent.

The identification of delay-causes is a focus in the latter phases of delay analysis, during causation analysis.

d. Delay Characterization is Independent of Responsibility

ALV's are considered "delays" independent of the responsibility for those variances. Thus an ALV can be contractor-caused or owner-caused, but it is still a delay. Similarly, the characterization of delays as 'excusable', 'compensable', 'concurrent' and 'paced' are attributes that are assigned well after the initial delay analysis starts by examining ALVs based on the causation analysis that has been performed after the schedule analysis is completed.

2. Identifying and Collecting Delays

a. Two Main Approaches to Identification & Collection

i. Cause-Based Approach: This approach starts with the collection of suspected causes of delays and then determining the effect they had on the baseline schedule and individual schedule updates. It is a 'causes in search of effects' approach. This is often used in the additive modeling methods. For example, an analysis may review the monthly reports, searching for issues that may have caused delays to the project.

ii. Effect-Based Approach: This approach is the opposite of the cause-based approach. It starts by compiling a set of ALV's and then identifies the causes of those variances. Specific documents that are associated with the time-frame, activity description, and amount of ALV's are reviewed to see if they could have created this variance. This approach is applied in the observational and the subtractive modeling methods. In the majority of the analysis scenarios, the effect-based approach is the more economical approach.

b. Criticality of the Delay

It is important for an analyst to not prejudge criticality, nor limit the collection process to only those delays perceived to affect the critical path, especially if the delays are being identified for a modeled method. In addition, a path that is near critical in one window maybe become critical in the next especially if delays are being extracted from the critical path. For example, in the Modeled / Subtractive / Single Simulation (MIP 3.8) and the Modeled / Subtractive / Multiple Base (MIP 3.9) methods, as delays are being stripped from the critical path, the path will "collapse" and the first near critical path will become critical. This is an iterative method and therefore, paths may collapse numerous times so that a path that originally has plenty of float becomes the critical path. The ultimate critical path quantification from the effect of each delay will eventually be determined in the modeling process. It is impossible for the analyst to know what the final critical path is until all of these delays have been added in (MIP's 3.6 and 3.7 or extracted out (MIP's 3.8 and 3.9).

Also, float consumption and ownership can be relevant where issues involve disruption, loss of productivity, and constructive acceleration regardless of the criticality of the activity.

3. Quantification of Delay Durations and Activity Level Variances

There are two fundamentally different methods for quantifying delay durations. They are the
variance method and the independent method.

a. Variance Method:

The variance method is a comparative method that determines the delay duration by computing the ALV between the as-built activity duration and the unimpacted or planned activity duration obtained from the baseline schedule, an updated schedule or other non-CPM sources such as a measured-mile analysis or some reasoned estimate. This method is purely mathematical in nature. Two figures (a planned and an actual) are subtracted from each other to compute the variance. These two figures may be dates, durations, or productivity measurements. Thus, the entire variance needs to be tied to one or more causes for the variance.

b. Independent Method:

In contrast, the independent method is not comparative. The delay duration is determined from project documentation that contemporaneously chronicled or otherwise recorded the occurrence of the delay or quantified the impact resulting from a delay event. Under this method, the answer to the question whether causation has been established or not depends on the type and content of the documentation that was used for the quantification.

For example, if the documentation consists of a daily diary entry that states that a specific activity was suspended for that specific day pending an investigation of a differing site condition, there is prima facie establishment of causation (one day of delay is clearly stated). But if the documentation is a letter stating that, "during the previous month many activities experienced extensive delays due to Owner-changes," further analysis to determine the delay duration and which activities were affected by the delaying events will be needed.

The example below is given to illustrate the difference between the variance and independent method: Suppose that the ALV for a specific activity is ten days. In the variance method, the entire ten days will be distributed among the responsible parties. However, in the independent method, the ALV is not even looked at in the beginning. Instead, the analyst researches project documentation to determine the delay amount. Therefore, if the project documentation only states that the activity was delayed three days by an event, the remaining seven days of the ALV will not be assigned to this delay and may not be assigned to the party responsible for this delay. If the documentation states the delay event was twelve days, the analyst will consider the delay to the activity was twelve days but since the ALV is ten, the other two days may have been made up via acceleration. Therefore, in the variance method, the analyst is guided to the delay amount by the amount of ALV. On the other hand, in the independent method, the analyst does not review the ALV, but relies on what is written in the documentation to make its determination of delay amount.

4. Cause of Variance

What caused the variances often does not appear on the schedule but must be investigated and researched using project documents, data and witness interviews. In researching, evaluating and modeling the cause-and-effect relationships, the analyst must be aware that these relationships are often successively linked into a chain of alternating causes and effects. In addition, an ALV may be created by more than one cause.

Causation is established primarily on the quality of documents available for the analyst at the time of the schedule analysis. Some documents are more reliable than others. Development of a document-type list and a reliability assessment for each document type are often the first steps prior to a detailed review of the record. This list is essential for two reasons. First, the analyst will become familiar with the types of documents that are available for review. Discussions with the project team concerning types of documents as well as the chronology of events will optimize the causation research process. For example, if the analyst is not aware that daily construction reports exist, and instead relies on monthly reports for determining causation, its conclusions of delay amount and impact may be very different.

5. Assigning or Assuming Variance Responsibility

When the forensic schedule analyst does not possess adequate information to make an independent determination of responsibility for the delay, the analyst may have to proceed with the analysis based on an assumption. Such assumptions should be noted and clearly stated as part of the final analysis product along with the basis of such assumption.

a. Contractor delay is any delay event caused by the contractor or those under its control, or the risk of which has been assigned solely to the contractor . Typical examples of contractor delay events include, but are not limited to, delays caused by rework resulting from poor workmanship, subcontractor delays, insufficient labor, management and coordination problems, failure to order necessary materials and failure to secure contractual approvals.

b. Owner delay is any delay event caused by the owner, or the risk of which has been assigned solely to the owner . Examples of owner-delay events include, but are not limited to, delays resulting from change orders, extended submittal review, directed suspension of work, delayed owner-furnished equipment, differing site conditions, and defective contract documents.

c. **Force majeure delay** is any delay event caused by something or someone other than the owner (including its agents) or the contractor (or its agents), or the risk of which has not been assigned solely to the owner or the contractor. Examples of force majeure delays include, but are not limited to, delays caused by acts of God, inclement weather, acts of war, extraordinary economic disruptions, strikes, and other events not foreseeable at the time of contract. Many contracts specifically define force majeure events. Although strictly not a ‘force majeure’ event, delays caused by parties external to the contract may also be classified under this category if there are no contractual risk assignment to the contractor or the owner for such delays

B. Recommended Protocol

1. Determine the delay identification and collection approach to be used.

2. Tabulate all sources of delay data and evaluate each for reliability. If the documentation sources have conflicting data, the analyst should use the source that is the most reliable and explain why the source used is considered the most reliable. 3. Identify the specific actual start date and actual finish date for each delay along with the scope of work that occurred on those dates and their significance in relation to the delay.

4. Correlate the delay event to the specific activity or activities in the schedule affected by the delay and determine if it affected the start of the activity or the duration of the activity.

5. Identify, tabulate, and quantify all significant activity-level variances. The significance of the ALV is done on a case by case basis, but the criteria for that significance and their bases should be noted.

6. Determine the criticality of those significant ALVs.

7. Determine the causation of those significant ALVs based on the correlation of delay event to activity as described in step number four.

8. Determine responsibility or proceed based on assumed allocation of responsibility

9. Quantify the claim portion of each ALV for which causation has been determined.

   a. If the delay is not a complete stoppage or not continuous throughout the entire period of the activity's duration, quantify the net delay duration during that time frame.

   b. For each delay issue, if applicable, distinguish the informational delay portion from the actual performance of disputed/extra work.

   c. For each discrete delay event, identify the activity ID number or numbers of the schedule activity or activities that were impacted by the delay.

C. Recommended Enhanced Protocol

1. Establish the activity coding structure for various attributes of delays, such as responsibility, issue grouping and documentation source so that different delay scenarios can be analyzed and relevant reports can be generated with minimal difficulty.

2. For each delay issue, if applicable, document and reconcile the claimed delay duration against any contract time extensions already received for that issue. The analyst needs to ensure that the entitlement quantification does not overlap or "double-dip" on pre-existing granted time extensions.

D. Special Procedures

1. Duration & Lag Variance Analysis

Prepare a table comparing the planned duration of a schedule activity to the actual duration and determine the cause for each significant variance.

Prepare a table comparing the planned controlling predecessor logic of the schedule activity to the actual controlling predecessor logic and determine the cause for each significant variance both in terms of change in type of logic and lag values.

3. METHOD IMPLEMENTATION

The intent of the Method Implementation Protocols (MIP) is to describe each forensic schedule analysis method identified in the Taxonomy and to provide guidance in implementing these methods. The user is reminded that the focus of this RP is on procedure as opposed to substance. Adopting a method and using the recommended procedures do not, on their own, assure soundness of substantive content.

The use of the Source Validation Protocols (SVP) discussed in Section 2 is integral to the implementation guidelines discussed here. Therefore a thorough understanding of the SVP is a prerequisite to the competent use of the MIP.

Method implementation protocols consist of the following:

3.1. Observational / Static / Gross (MIP 3.1)
3.2. Observational / Static / Periodic (MIP 3.2)
3.3. Observational / Dynamic / Contemporaneous As-Is (MIP 3.3)
3.4. Observational / Dynamic / Contemporaneous Split (MIP 3.4)
3.5. Observational / Dynamic / Modified or Recreated (MIP 3.5)
3.6. Modeled / Additive / Single Base (MIP 3.6)
3.7. Modeled / Additive / Multiple Base (MIP 3.7)
3.8. Modeled / Subtractive / Single Simulation (MIP 3.8)
3.9. Modeled / Subtractive / Multiple Base (MIP 3.9)

3.1. Observational / Static / Gross (MIP 3.1)

A. Description

MIP 3.1 is an observational technique that compares the baseline or other planned schedule to the as-built schedule or a schedule update that reflects progress.

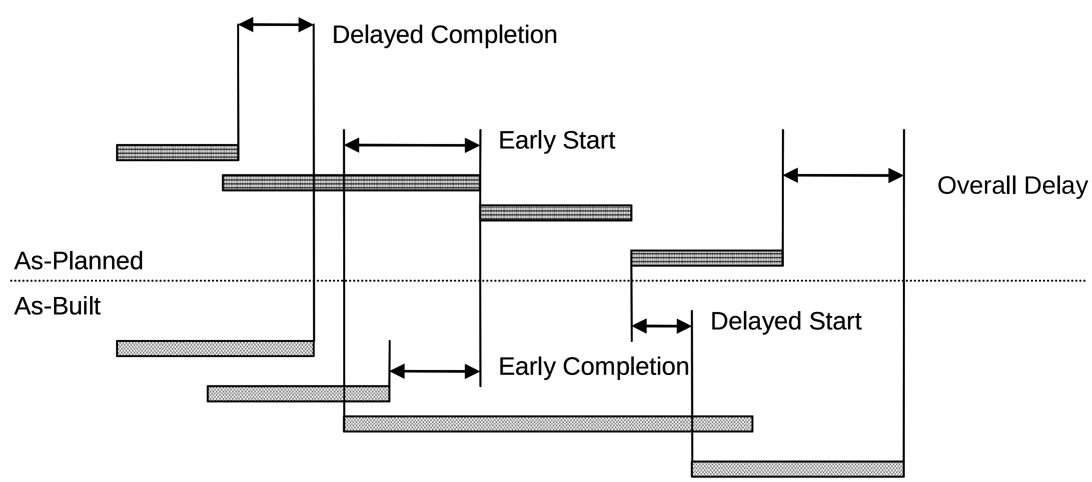


Figure 3 – Observational, Static, Gross Analysis Method Graphic Example

In its simplest application, the method does not involve any *explicit* use of CPM logic and can simply be an observational study of start and finish dates of various activities. It can be performed using a simple graphic comparison of the as-planned schedule to the as-built

schedule. A more sophisticated implementation compares the dates and the relative sequences of the activities and tabulates the differences in activity duration, and logic ties and seeks to determine the causes and explain the significance of each variance. In its most sophisticated application, it can identify on a daily basis the most delayed activities and candidates for the as-built critical path.
MIP 3.1 is classified as a static logic method because it primarily relies on the single set of CPM logic underlying the baseline or other planned schedule. The method is classified as gross as opposed to periodic because the analysis is performed on the entire project against a single baseline or other planned schedule rather than in periodic segments.

B. Common Names

1. As-planned vs. as-built
2. AP vs. AB
3. Planned vs. actual
4. As-planned vs. update

C. Recommended Source Validation Protocols

1. Implement SVP 2.1 (baseline validation) and,
2. Implement SVP 2.2 (as-built validation) or,
3. Implement SVP 2.3 (update validation) and,
4. Implement SVP 2.4 (delay ID & quantification)

D. Enhanced Source Validation Protocols

[Not used.]

E. Recommended Implementation Protocols

The application of this methodology involves the sequential comparison of individual activities' planned start and finish dates with actual start and finish dates. Through this comparison, a detailed summary of the delays and/or accelerations of activities can be identified. Generally, it is best to compare the LATE planned dates from a CPM schedule, rather than the early dates. While contractors usually intend to perform their work in accordance with the early dates, delay to an activity cannot be measured until the activity is actually delayed–is later than the planned late dates. The basic steps in the analysis are as follows:

1. Identify the baseline or other schedule that will form the as-planned schedule. Ideally, this schedule reflects a schedule that has been approved or accepted by both parties and reflects the full scope of the work, includes proper logic from the start of the project through completion, and reflects neither progress nor post-commencement mitigations of delay. This schedule is usually a CPM model, so that even without functioning CPM logic and modeling, the original planned logic should be used in analysis and interpretation. Alternatively, a simple comparison can be performed using graphic time-scaled diagrams. In this situation, no explicit schedule logic is evident, although the sequence and timing will imply certain logical connections.

2. The comparison progresses from the earliest activities' planned dates to later dates. Generally, this comparison sequence should follow the logic in the original as-planned schedule. Thus, at least until the first significant delays, the focus will be on the as-planned critical and near-critical paths.

3. The analysis should advance through the comparison by identifying for each activity: (a) delayed starts, (b) extended durations, and (c) delayed finishes. Since the as-built analysis is performed using a 7-day calendar, it is important that all durations be in calendar days. In this manner, it is possible to identify where the most significant delays occurred, where there were mitigations of delay through implementation of out-of-sequence logic and possible accelerations through shorter than planned durations.

4. Arithmetic calculations performed at the start and completion of each as-built activity provide a detailed view of the relative delay of every as-built activity. The most delayed series of activities can be ascertained using this method and can often be used as a starting point for identifying the as-built critical path. Expert judgment is required to separate the as-built critical path (based on industry experience and contemporaneous evidence as discussed in Subsection 4.3.C) from the various set of most delayed activities at any particular time.

5. Simultaneous delays, whether they are pacing delays (see Subsection 4.2.B) or concurrent delays (see Subsection 4.2.A), should be identified and confirmed as being on the critical path.

6. As the analysis continues and advances through the as-planned schedule, it is likely that it will become less accurate since contemporaneous adjustments to the contractor's plan will supersede the original logic. For this reason, particular care must be exercised during the analysis of the later stages of the project.

7. Extended durations for any activity should be examined for the cause. This will determine the cause of the delays along the critical path.

8. Similarly, any duration with shorter than planned durations may indicate reductions in work scope or acceleration by the contractor.

9. If time extensions have been granted, they should be considered both at the time they were granted and at the end of the analysis. Time extensions should be considered when evaluation of the reasons for delayed performance is identified through the comparison as well as identification of the as-built critical path. Time extensions will change the overall delay to the project and may therefore override apparent delays to specific activities.

If the baseline schedule has both early and late dates, the analysis should be performed using late dates unless a review of the late dates reveal that the logic associated with the late dates is significantly different than the logic of the early dates. In this situation, the analysis should be performed using early dates with the understanding that adjustments for available float may need to be considered. A schedule with logic that is incomplete or significantly different from the logic associated with the early dates should be considered for correction in accordance with Subsection 2.1.B.

The minimum implementation of this method is applicable only to relatively simple cases and should not be used for long duration cases or where there are significant changes between the original planned work scope and the final as-built scope. For the purpose of this MIP, a 'simple case' is defined as one in which there is a single clearly defined chain of activities on the longest path that stayed as the longest path throughout the performance of the project.

F. Enhanced Implementation Protocols

1. Daily Delay Measure

The as-planned vs. as-built methodology can be used in more complicated cases if the data is available. Since the basic implementation protocol is applicable only for very simple cases, this more advanced method should be used if possible. However, even this more enhanced implementation is useful only for simple projects where the sequence of work did not vary significantly from the baseline schedule.

a. The as-built should be a fully progressed baseline schedule allowing for a one-to-one comparison of each schedule activity. This is essential as activity descriptions and ID numbers often change as the project advances.

b. On larger schedules and projects that are active for long periods of time, it is often desirable to use a database comparison between actual dates determined from the as-built analysis with the LATE planned dates. This comparison will allow the selection of the more significant activities for graphical comparison. Prepare a table comparing the planned duration or a schedule activity to the actual duration and determine the cause for each significant variance.

c. Prepare a table comparing the planned controlling predecessor logic of schedule activity to the actual controlling predecessor logic and determine the cause for each significant variance.

d. If an edited baseline schedule was used, the analysis should proceed using both the unaltered baseline as well as the modified baseline. A comparison between the two sets of results will assist the analysis in identifying the likely and realistic progress of the job.

e. Arithmetic calculations performed on a daily basis can provide significantly more accurate information if the as-built data is available at the appropriate level of detail. This method is called Daily Delay Measure (DDM). DDM is an enhanced variation for the identification of activities that are candidates for critical and near critical paths. DDM compares late start and finish dates with as-built start and finish dates.

- It can be done on a daily, weekly, or any other periodic basis. By depicting the number of days a schedule activity is ahead or behind the planned late dates, a determination of any point of the status of any schedule activity is possible.
- While the comparison can be made between the early start/finish dates and the actual dates, it is better to compare late start/finish dates with actual dates. By using late dates, any delay indicated by the comparison is a true delay rather than consumption of float. As a result of that exercise, any float associated with the duration of a schedule activity is excluded. Activities that have float (and accordingly are not on the as-planned critical path) will generally not appear to have been delayed during the early stages of analysis, since they will appear to be "ahead" of schedule because of their float. As the analysis progresses through a project's performance however, the activities that initially had float, if they were delayed for a duration in excess of the value of that float, can become critical, thus overtaking one or more of those activities originally on the project's as-planned critical path. While late dates are preferred in performing the analysis, in

some CPM schedules, late dates do not represent a consistent or practical plan for execution of the work even if the early dates do. In these cases, it is better to use early dates.

- The DDM can also identify possible changes in the as-built critical path if the analysis is done on a frequent, possible daily basis, even within the actual duration of activities. In this case there are several alternative assumptions that can be made to identify progress within an activity duration: (1) if accurate progress data is available on a regular basis, this regular progress can be used (realistically this is rare in most construction projects); (2) progress can be assumed to advance at an equal rate each period, for example, a 10-day activity would be assumed to advance 10 percent each day; or (3) a different progress rate, perhaps conforming to a more typical bell-curve distribution.

G. Identification of Critical and Near-Critical Paths

In this method, the emphasis should be on the as-built critical path as opposed to the baseline critical path. Since this methodology does not use a computational CPM, the methodology relies more extensively on expert evaluation.

- Identify and understand all related contractual language.

- From the fully populated baseline schedule, identify the calculated critical path of the baseline using the longest path and the lowest total float concept of the validated baseline.

- From the fully populated as-built schedule, identify the near-critical path using the procedure in Subsection 4.3.C. for identifying the as-built critical path.

- Confirm and cross check these results by tracing the delays through the as-planned critical path and near critical paths based on late as-planned dates.

- Identify the most delayed activities at every measuring point.

- Review the planned logic and evaluate any likely changes based on contemporaneous evidence.

H. Identification & Quantification of Concurrent Delays & Pacing

- Identify and understand all related contractual language.

- Determine whether literal or functional concurrency theory is to be used.

- If applicable, determine the near-critical threshold (see Subsection 4.3.).

- If applicable, determine the frequency, duration, and placement of the analysis intervals.

- Determine whether there are two simultaneous delays to activities on the critical path, or two simultaneous causes of delay to a single activity on the as-built critical path.

- Determine the day each delay commenced or period within which each commenced.

- Determine the contractually responsible party for each delay by the contractor or owner at issue.

- For each delay event, distinguish the cause from the effect of delay.

- Identify and explain all relative delayed starts and extended duration of activities that are critical or near-critical.

- For each suspected pacing delay event, identify the parent delay(s) and establish the order of precedence between the parent delay and the pacing delay.

- For each suspected pacing delay event, evaluate whether enough resources could have been realistically employed to perform the paced activity within its original planned duration.

I. Determination & Quantification of Excusable and Compensable Delay

Identify and understand all contractual language related to delay apportionment and determine whether the contractual language would override any determination of excusability and compensability based on findings resulting from analyses performed under this RP.

Absent such overriding language, use the following procedure:

1. Excusable & Compensable Delay (ECD)

Each incremental delay along the as-built critical path should be independently quantified and the cause of the delay identified. The net Excusable & Compensable Delay (ECD) is the sum of the individual delays that: 1) were the responsibility of the owner, and 2) delayed the completion date of the project, and 3) were not concurrent with delays which were the responsibility of the contractor or *force majeure* events.

2. Excusable & Non-compensable Delay (END)

Each incremental delay along the as-built critical path should be independently quantified and the cause of the delay identified. The net Excusable & Non-compensable Delay (END) is the sum of the individual owner-caused or relevant third-party caused delays that: 1) were *force majeure* events or were concurrent with contractor-responsible delays or *force majeure* events, and 2) delayed the completion date of the project, and 3) were not the responsibility of the contractor.

J. Identification & Quantification of Mitigation / Constructive Acceleration

Observational / static analysis methods can note differences in logic but cannot directly quantify net critical path impact. However, there may be evidence of reduced individual activity duration, which when coupled with detailed records of increased man-hours, would serve as adequate proof of acceleration. Note that the acceleration would be evident in both critical path and non-critical path activities.

K. Specific Implementation Procedures & Enhancements

[Not Used]

L. Summary of Considerations In Using the Minimum Protocol

- Suitable for analyzing short projects with minimal logic changes.

- Can be performed in a manner that is easy to understand and simple to present.

- Technically simple to perform compared to other MIP's.
- Can be performed with very rudimentary schedules and as-built data.
- As-built activities must be closely correlated with as-planned activities.
- As-built data used must be accurate and validated.
- Does not, by itself, identify the as-built critical path.

M. Caveats In Using the Minimum Protocol / Conditions Requiring Enhanced Protocols

- Not suitable for project durations extending into multiple years.
- Not suitable for projects built in a manner significantly different than planned. The rate of error increases as the incidence of change increases.
- Not suitable for complicated projects with multiple critical paths.
- Does not consider the possibility of critical path shifts either within periods or across the project.
- Susceptible to unintentional or intentional manipulation by choice of as-built data that is incorporated into schedule.
- May fail to identify all critical delays or time extensions, and typically does not adequately consider concurrency and pacing issues.
- Does not consider that changes to original baseline schedule may have been the actual cause of delay instead of the identified delay issues
- Typically fails to consider chronological order of delays or reconcile periodic planned critical path shifts with the as-built critical path
- Not suited for clearly demonstrating acceleration

3.2. Observational / Static / Periodic (MIP 3.2)

A. Description

Like MIP 3.1, 3.2 is an observational technique that compares the baseline or other planned schedule to the as-built schedule or a schedule update that reflects progress. But, this method analyzes the project in multiple segments rather than in one whole continuum. Because this is essentially an enhancement of MIP 3.1, as a practical matter, the implementation of MIP 3.2 requires that prerequisites for MIP 3.1 be implemented first.

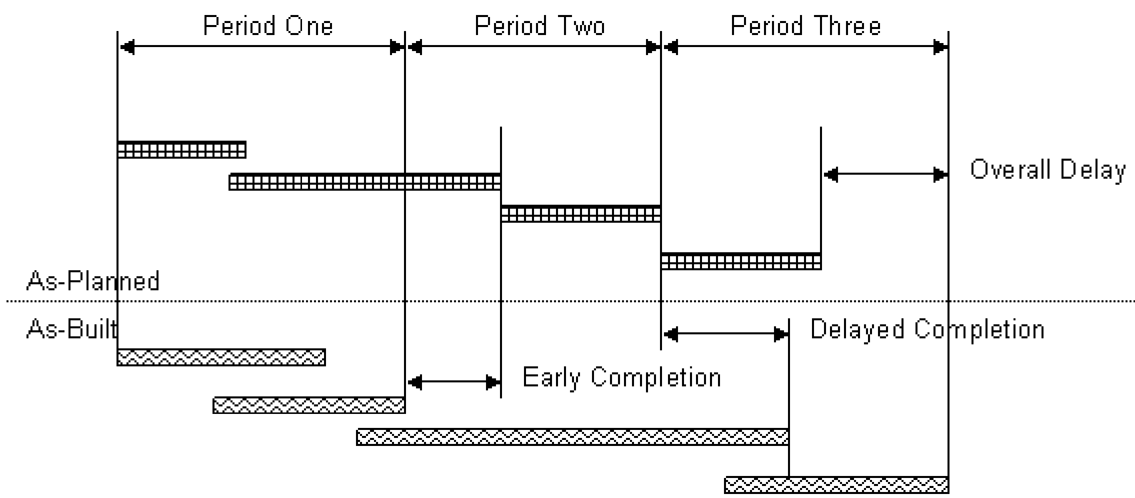


Figure 4 – Observational, Static, Periodic Method Graphic Example

In its range of implementation from simple to sophisticated, it shares the characteristics of MIP 3.1. In its simplest application, the method does not involve any *explicit* use of CPM logic and can be simply an observational study of start and finish dates of various activities. It can be performed using a simple graphic comparison of the as-planned schedule to the as-built schedule. A more sophisticated implementation compares the dates and the relative sequences of the activities, tabulates the differences in activity duration and logic ties, seeks to determine the causes, and explains the significance of each variance. In its most sophisticated application, it can identify on a daily basis the most delayed activities and candidates for the as-built critical path.

The advantage of performing this analysis in two or more time periods is that the identification of delays or accelerations can be more precisely identified to particular events. Generally the more time periods, the more closely related the analysis is to the events that actually occurred. The fact that the analysis is segmented into periods does not significantly increase or decrease the technical accuracy of this method when compared to MIP 3.1 because the comparison remains between the as-built and baseline or original as-planned schedule. However, the segmentation is useful in enhancing the organization of the analysis process and enables prioritization. It also may add to the effectiveness of the presentation of the analysis.

MIP 3.2 is classified as a static logic method because it primarily relies on the single set of CPM logic underlying the baseline schedule or other planned schedule. Note that a similar method as described in MIP 3.3 is classified as a dynamic logic method because that method uses a series of updates schedule with logic that may be different from the baseline and from each other. MIP 3.2 is distinguished from MIP 3.3 in that while the analysis is performed in segments, they are segments of the as-planned and as-built without reference to schedule updates that are contemporaneous to those segments.

The method is classified as periodic because the analysis is performed in periodic segments rather than in one continuous project period.

B. Common Names

1. As-planned vs. as-built

2. AP vs. AB

3. Planned vs. actual

4. As-planned vs. update

5. Window analysis

6. Windows analysis

C. Recommended Source Validation Protocols

1. Implement SVP 2.1 (baseline validation) and,

2. Implement SVP 2.2 (as-built validation) or,

3. Implement SVP 2.3 (update validation) and,

4. Implement SVP 2.4 (delay ID & quantification)

D. Enhanced Source Validation Protocols

[Not used.]

E. Recommended Implementation Protocols

The procedures below are essentially those of MIP 3.1, but are applied only for a specific time period which is less than the overall duration of the project. Selection of the time periods should follow Subsection 3.2.A. In this method however, the selection is primarily made for clarity of conclusions, not for greater accuracy of analysis.

The results of this analysis are summed at the end of each time analysis period. The application of this methodology involves the sequential comparison of individual activities' planned start and finish dates with actual start and finish dates. Through this comparison, a detailed summary of the delays and/or accelerations of activities can be identified. Generally, it is best to compare the LATE planned dates from a CPM schedule rather than the early dates. While contractors usually intend to perform their work in accordance with the early dates, delay to an activity cannot be measured until the activity is actually delayed–is later than the planned late dates. The basic steps in the analysis are as follows:

1. Identify the baseline or other schedule that will form the as-planned schedule. Ideally, this schedule reflects a schedule that has been approved or accepted by both parties and reflects the full scope of the work, includes proper logic from the start of the project

through completion, and reflects neither progress nor post-commencement mitigations of delay. This schedule is usually a CPM model, so that even without functioning CPM logic and modeling, the original planned logic should be used in analysis and interpretation. Alternatively, a simple comparison can be performed using graphic time-scaled diagrams. In this situation, no explicit schedule logic is evident, although the sequence and timing will imply certain logical connections.

2. The comparison progresses from the earliest activity planned dates to later dates. Generally, this comparison sequence should follow the logic in the original as-planned schedule. Thus, at least until the first significant delays, the focus will be on the as-planned critical and near-critical paths.

3. The analysis should advance through the comparison by identifying for each activity: (a) delayed starts, (b) extended durations, and (c) delayed finishes. Since the as-built analysis is performed using a 7-day calendar, it is important that all durations be in calendar days. In this manner, it is possible to identify where the most significant delays occurred, in which there were mitigations of delay through implementation of out-of-sequence logic, and possible accelerations through shorter than planned durations.

4. Arithmetic calculations performed at the start and completion of each as-built activity provide a detailed view of the relative delay of every as-built activity. The most delayed series of activities can be ascertained using this method and can often be used as a starting point for identifying the as-built critical path. Expert judgment is required to identify the as-built critical path, based on industry experience and contemporaneous evidence as discussed in Subsection 4.3.C, from the various set of the most delayed activities at any particular time.

5. Simultaneous delays, whether they are pacing delays (see Subsection 4.2.B), or concurrent delays (see Subsection 4.2.A), should be identified and confirmed as being on the critical path.

6. As the analysis continues and advances through the as-planned schedule, it is likely that it will become less accurate since contemporaneous adjustments to the contractor’s plan will supersede the original logic. For this reason, particular care must be exercised during the analysis of the later stages of the project.

7. Extended durations for any activity should be examined for the cause. This will determine the cause of the delays along the critical path.

8. Similarly, any activities with shorter than planned durations may indicate reductions in work scope or acceleration by the contractor.

9. If time extensions have been granted, they should be considered both at the time they were granted and at the end of the analysis. Time extensions should be considered when evaluating the reasons for delayed performance identified through the comparison as well as identification of the as-built critical path. Time extensions will change the overall delay to the project and may therefore override apparent delays to specific activities.

10. Prepare a table that summarizes the variances quantified for each analysis period and reconcile the total to the result that would be obtained by a competent implementation of MIP3.1. This is intended to eliminate the possibility of skewing the result of the analysis through the use of variable periods.

If the baseline schedule has both early and late dates, the analysis should be performed using late dates unless a review of the late dates reveal that the logic associated with the late dates is significantly different than the logic of the early dates. In this situation, the analysis should be performed using early dates with the understanding that adjustments for available float may need to be considered. A schedule with logic that is incomplete or significantly different from the logic associated with the early dates should be considered for correction is accordance with Subsection 2.1.B.

The minimum implementation of this method is applicable only to relatively simple cases and should not be used for long duration cases or where there are significant changes between the original planned work scope and the final as-built scope. For the purpose of this MIP, a 'simple case' is defined as one in which there is a single clearly defined chain of activities on the longest path that stayed as the longest path throughout the performance of the project.

F. Enhanced Implementation Protocols

1. Daily Delay Measure

The as-planned vs. as-built methodology can be used in more complicated cases if the data is available. Since the basic implementation protocol is applicable only for very simple cases, this more advanced method should be used if possible. However, even this more enhanced implementation is useful only for simple projects where the sequence of work did not vary significantly from the baseline schedule.

a. The as-built should be a fully progressed baseline schedule allowing for a one-to-one comparison of each schedule activity. This is essential as activity descriptions and ID numbers often change as the project advances.

b. On larger schedules and projects that are active for long periods of time, it is often desirable to use a database comparison between actual dates determined from the as-built analysis with the LATE planned dates. This comparison will allow the selection of the more significant activities for graphical comparison. Prepare a table comparing the planned duration or a schedule activity to the actual duration and determine the cause for each significant variance.

c. Prepare a table comparing the planned controlling predecessor logic of schedule activity to the actual controlling predecessor logic and determine the cause for each significant variance.

d. If an edited baseline schedule was used, the analysis should proceed using both the unaltered baseline as well as the modified baseline. A comparison between the two sets of results will assist the analysis in identifying the likely and realistic progress of the job.

e. Arithmetic calculations performed on a daily basis can provide significantly more accurate information if the as-built data is available at the appropriate level of detail. This method is called Daily Delay Measure (DDM). DDM is an enhanced variation for the identification of activities that are candidates for critical and near critical paths. DDM compares late start and finish dates with as-built start and finish dates.

    - It can be done on a daily, weekly, or any other periodic basis. By depicting the number of days a schedule activity is ahead or behind the planned late dates, a determination at any point of the status of any schedule activity is possible.

- While the comparison can be made between the early start/finish dates and the actual dates, it is better to compare late start/finish dates with actual dates. By using late dates, any delay indicated by the comparison is a true delay rather than consumption of float. As a result of that exercise, any float associated with the duration of a schedule activity is excluded. Activities that have float (and accordingly are not on the as-planned critical path) will generally not appear to have been delayed during the early stages of analysis, since they will appear to be "ahead" of schedule because of their float. As the analysis progresses through a project's performance however, the activities that initially had float, if they were delayed for duration in excess of the value of that float, can become critical, thus overtaking one or more of those activities originally on the project's as-planned critical path. While late dates are preferred in performing the analysis, in some CPM schedules, late dates do not represent a consistent or practical plan for execution of the work even if the early dates do. In these cases, it is better to use early dates, taking into account the float values.

- The DDM can also identify possible changes in the as-built critical path if the analysis is done on a frequent, possibly daily basis, even within the actual duration of activities. In this case, there are several alternative assumptions that can be made to identify progress within an activity duration: (1) if accurate progress data is available on a regular basis, this regular progress can be used (realistically this is very rare in most construction projects); (2) progress can be assumed to advance at an equal rate each period, for example a 10-day activity would be assumed to advance 10 percent each day; or (3) a different progress rate, perhaps conforming to a more typical bell-curve distribution.

G. Identification of Critical & Near-Critical Paths

In this method, the emphasis should be on the as-built critical path as opposed to the as-planned critical path. Since this methodology does not use a computational CPM, the methodology relies more extensively on expert evaluation.

- Identify and understand all related contractual language.

- From the fully populated baseline schedule, identify the calculated critical path of the as-planned using the longest path and the lowest total float concept of the validated as-planned schedule.

- From the fully populated as-built schedule, identify the near-critical path using the procedure in Subsection 4.3.C. for identifying the as-built critical path.

- Confirm and cross check these results by tracing the delays through the as-planned critical path and near critical paths based on late as-planned dates.

- Identify the most delayed activities at every measuring point.

- Review the planned logic and evaluate any likely changes based on contemporaneous evidence.

H. Identification & Quantification of Concurrent Delays & Pacing

- Identify and understand all related contractual language.

- Determine whether literal or functional concurrency theory is to be used.

- If applicable, determine the near-critical threshold (see Subsection 4.3.).

- If applicable, determine the frequency, duration, and placement of the analysis intervals.

- Determine whether there are two simultaneous delays to activities on the critical path or two simultaneous causes of delay to a single activity on the as-built critical path.

- Determine the day each delay commenced or period within which each commenced.
- Determine the contractually responsible party for each delay by the contractor or owner at issue.

- For each delay event, distinguish the cause from the effect of delay.

- Identify and explain all relative delayed starts and extended duration of activities that are critical or near-critical.

- For each suspected pacing delay event, identify the parent delay(s) and establish the order of precedence between the parent delay and the pacing delay.

- For each suspected pacing delay event, evaluate whether enough resources could have been realistically employed to perform the paced activity within its original planned duration.

I. Determination & Quantification of Excusable and Compensable Delay

Identify and understand all contractual language related to delay apportionment and determine whether the contractual language would override any determination of excusability and compensability based on findings resulting from analyses performed under this RP.

Absent such overriding language, use the following procedure.

1. Excusable & Compensable Delay (ECD)

Each incremental delay along the as-built critical path should be independently quantified and the cause of the delay identified. The net Excusable & Compensable Delay (ECD) is the sum of the individual delays that: 1) were the responsibility of the owner, and 2) delayed the completion date of the project, and 3) were not concurrent with delays which were the responsibility of the contractor or *force majeure* events.

2. Excusable & Non-compensable Delay (END)

Each incremental delay along the as-built critical path should be independently quantified and the cause of the delay identified. The net Excusable & Non-compensable Delay (END) is the sum of the individual owner-caused delays that: 1) were *force majeure* events or were concurrent with contractor-responsible delays or *force majeure* events, and 2) delayed the completion date of the project, and 3) were not the responsibility of the contractor.

J. Identification & Quantification of Mitigation / Constructive Acceleration

Observational / Static analysis methods can note differences in logic but cannot directly quantify net critical path impact. However, there may be evidence of reduced individual activity duration, which when coupled with detailed records of increased man-hours, would serve as adequate proof of acceleration. Note that the acceleration would be evident in both critical path and non-critical path activities.

K. Specific Implementation Procedures & Enhancements

1. Fixed Periods

The analysis periods are of virtually identical duration and may coincide with regular schedule update periods.

2. Variable Periods

The analysis periods are of varying durations and are characterized by their different natures such as the type of work being performed, the types of delaying influences, significant events, changes to the critical path, revised baseline schedules, and/or the operative contractual schedule under which the work was being performed.

Fixed periods have the advantage of providing regular measurements and thus make it easier to track progress through the project. However, variable periods identified by major events on the project are often more useful since they will relate status of the delay to a specific known event.

L. Summary of Considerations In Using the Minimum Protocol

- Allows for logical segmenting of relatively longer project durations than MIP 3.1
- Suitable for analyzing short projects with minimal logic changes.
- Can be performed in a manner that is easy to understand and simple to present.
- Technically simple to perform compared to other MIP's, other than MIP 3.1. However it is still relatively time consuming when implemented correctly.
- Can be performed with very rudimentary schedules and as-built data.
- As-built activities must be closely correlated with as-planned activities.
- As-built data used must be accurate and validated.
- Does not, by itself, identify the as-built critical path.

M. Caveats In Using the Minimum Protocol / Conditions Requiring Enhanced Protocols

- Provides illusion of greater detail and accuracy compared to MIP 3.1 where none exists since it still does not consider the possibility of critical path shifts either within periods or across the project.
- Does not use the contemporaneous as-planned update predictions of critical paths
- The choice of variable periods may be abused to skew the results of the analysis.
- Not suitable for project durations extending into multiple dozens of update periods.

- Not suitable for projects built in a manner significantly different than planned. The rate of error increases as the incidence of change increases.
- Not suitable for complicated projects with multiple critical paths.
- Susceptible to unintentional or intentional manipulation by choice of as-built data that is incorporated into schedule.
- May fail to identify all critical delays or time extensions, and typically does not adequately consider concurrency and pacing issues.
- Does not consider that changes to original baseline schedule may have been the actual cause of delay instead of the identified delay issues
- Typically fails to consider chronological order of delays
- Typically fails to reconcile periodic planned critical path shifts with the as-built critical path
- Not suited for clearly demonstrating acceleration due to reliance on original as-planned logic only

3.3. Observational / Dynamic / Contemporaneous As-Is (MIP 3.3)

A. Description

MIP 3.3 is a retrospective technique that uses the project schedule updates to quantify the loss or gain of time along a logic path and identify the causes. Although this method is a retrospective technique, it relies on the forward-looking calculations made at the time the updates were prepared. That is, it primarily uses the information to the right of the updates' data dates.

MIP 3.3 is an observational technique since it does not involve the insertion or deletion of delays but instead is based on observing the behavior of the network from update to update and measuring schedule variances based on essentially unaltered, existing schedule logic.

Because the method uses schedule updates whose logic may have changed from the previous updates as well as from the baseline, it is considered a dynamic logic method.

It is labeled contemporaneous because the updates it relies on were prepared contemporaneously with the project execution as opposed to reconstructed after-the-fact as in MIP 3.5.

Finally, the 'as-is' label distinguishes this method from MIP 3.4 by the fact that the updates are evaluated almost completely untouched or 'as is'.

While rare, it is possible that no non-progress revisions were made in the contemporaneous updates. In this situation, this method should yield a result similar to a static logic method (MIP 3.1 and 3.2) since the initial baseline logic is in place for the entire project

B. Common Names

1. Contemporaneous period analysis
2. Contemporaneous project analysis
3. Observational CPA
4. Update analysis
5. Month-to-month
6. Window analysis
7. Windows analysis

C. Recommended Source Validation Protocols

1. Implement SVP 2.1 (baseline validation) and,
2. Implement SVP 2.3 (update validation)

D. Enhanced Source Validation Protocols

1. Implement SVP 2.2 (as-built validation)
2. Implement SVP 2.4 (identification of delay events)

E. Recommended Implementation Protocols

1. Recognize all contract time extensions granted.

2. Identify the critical path activity that will be used to track the loss or gain of time for the overall network.

3. Determine whether evaluations will be done on all periods or grouped periods as described in Subsection 3.3.K.

4. Not every update needs to be used, but accuracy tends to be reduced if multiple-month update periods are utilized.

5. Separately identify activities that will be used to track intra-network time losses and gains, such as interim milestones.

6. Compare the update at the start of the analysis period to the update at the end of the analysis period.

7. Use the longest path and the least float criteria to identify the controlling chain of activities.

8. Identify changes (gained or lost time) in overall Project completion date, and if necessary, in interim milestone completion dates.

9.

10. Identify start and finish variances of critical and near-critical activities in the analysis period.

11. Indentify all changes and/or revisions to logic, durations, and/or progress that were made during analysis period.

12. Identify responsibility for delays and gains during analysis period.

13. Continue with implementation until all periods are complete

14. Sum the net gains and losses for each period to arrive at an overall impact to the project. The sum of the net impacts must be equal to difference between the first schedule update and last schedule update used in the evaluation.

F. Enhanced Implementation Protocols

1. Daily Progress Method

The application of this methodology involves identifying the delay or savings in time attributable to the project's progress between the updates by chronologically tracking progress along the critical path on a unit basis (typically the smallest planning unit used in executing the project, for example, daily), by comparing the planned timing of the activities in the first update to their actual progress as depicted in the second, and identifying the resulting effect of the project's progress. The following steps outline the application of this methodology:

a. Identify the consecutive schedules that will be used to measure the delay or savings in time. For example, update No. 1 and update No. 2.

b. Using a copy of the first update, insert the progress made on day 1 of the update period, as depicted in the second update, and re-status the progressed update with a data date of the next calendar day.

c. Compare the critical paths of the first update and the progressed update to identify the activity(ies) whose progress or lack of progress affected the project's milestones.

d. Separately measure the effect of the responsible critical activity(ies) to the project milestones. In doing so, the analyst should separately identify critical activity(ies) that cause delay and other critical activities that may show out-of-sequence progress resulting in a savings in time to the project milestones.

e. Repeat this procedure of inserting the project's progress on a daily basis for every calendar day between the updates, while identifying and measuring the effect of progress on the critical paths of consecutive calendar days until reaching the data date of the second update.

f. This step concludes with the creation of a totally-progressed version of the first update, with the second update's data date, that contains all of the progress contained in the second update and that depicts the status of the project before the development of the second update.

The distribution of progress to activities that made progress between the updates can determine whether an activity becomes critical and potentially delays the project. For example, assume an activity started before the update period, made five workdays of progress during the update period, and was not completed during the update period. If there are no contemporaneous documents to identify when those five workdays of progress occurred, then the analyst has to decide when and how to depict the work occurring between the updates. The analyst could assume that the progress occurred within the first available five workdays of the period, or the last available workdays of the period, or in some other manner between the updates. Regardless of which method is chosen to distribute progress between the updates, the analyst should consistently apply the chosen method throughout the entire analysis and be able to explain why the method was chosen.

Upon completion of these steps, the analyst will be able to specifically identify the activities that were responsible for the delay or savings in time to the project's milestones during the update period and assign the resultant delay or savings to those same activities caused by the progress made between the updates. Additionally, by tracking the progress along the critical path between the updates the analyst will be able to identify shifts in the critical path.

This process is performed between all consecutive updates throughout the entire project duration.

G. Identification of Critical & Near-Critical Paths

- Identify and understand all related contractual language.

- Identify the negative float theory being used by the opposing analyst.

- For each analysis interval, identify the calculated critical path using the longest path and the lowest total float concept of the validated update(s) corresponding to the analysis interval.

- The near-critical activity-set in each analysis interval is the one that yields the most number of activities using one of the following methods:
  - lowest float value in the update PLUS the average duration of all discrete delay events contained in whole or in part inside the analysis interval, or
  - lowest float value in the update PLUS duration of the analysis interval.

H. Identification & Quantification of Concurrent Delays & Pacing

- Determine whether compensable delay by contractor or owner is at issue.
- Identify and understand all related contractual language.
- For each delay event, distinguish the cause from the effect of delay.
- Determine whether literal or functional concurrency theory is to be used.
- If applicable, determine the near-critical threshold (see Subsection 4.3.).
- If applicable, determine the frequency, duration, and placement of the analysis intervals.
- For each analysis interval, identify the critical path(s) and the near-critical path(s) and explain all relative delayed starts and extended duration of activities that occurred in the previous analysis interval on the same chains of activities.
- For each suspected pacing delay event, identify the parent delay(s) and establish the order of precedence between the parent delay and the pacing delay.
- For each suspected pacing delay event, evaluate whether enough resources could have been realistically employed to perform the paced activity within its original planned duration.

I. Determination & Quantification of Excusable & Compensable Delay

Identify and understand all contractual language related to delay apportionment and determine whether the contractual language would override any determination of excusability and compensability based on findings resulting from analyses performed under this RP.

Absent contract language or other agreements, use the following procedure to determine the net total delay apportionment:

1. Non-Excusable & Non-Compensable Delay (NND)

a. For each period analyzed, determine the longest-path delay attributable to events that are contractor-caused that occurred between the current data date and the last data date.

b. For each period analyzed, determine the longest-path gains attributable to contractor-initiated schedule mitigation that was actually implemented, and then add the resulting values together.

c. Make adjustment for concurrent delays due to owner-caused and *force majeure*-caused events using the selected concurrency analysis method.

2. Excusable & Compensable Delay (ECD)

a. For each period analyzed, determine the longest-path delay attributable to events that are owner-caused that occurred between the current data date and the last data date.

b. For each period analyzed, determine the longest-path gains attributable to owner-initiated schedule mitigation that was actually implemented, and then add the resulting values together.

c. Make adjustment for concurrent delays due to contractor-caused and *force majeure*-caused events using the selected concurrency analysis method.

3. Excusable & Non-Compensable Delay (END)

a. Total network delay less total NND less total ECD is the total END.

J. Identification & Quantification of Mitigation / Constructive Acceleration

The observational / dynamic analysis methods are especially well-suited for identifying and quantifying acceleration and delay mitigation through the use of logic changes. These methods allow the analyst to not only quantify the acceleration, but also determine whether the acceleration was achieved by current, actually implemented measures, or by logic changes representing promise of future acceleration.

With MIP 3.3, acceleration or delay mitigation is identified by comparing the completion date of the longest path of the previous period with that of the current period. A current date that is earlier than the previous date suggests acceleration. However, note that the value is a net number potentially representing both slippage and gain, where the gain was greater than the slippage. Thus a detailed examination of the longest path and the near-longest path surrounding the data date is necessary along with the examination of the logic changes between the last and the current periods along those paths is necessary for a competent identification and quantification of acceleration and delay mitigation.

In order to determine whether the promised future acceleration was actually implemented, it will be necessary to compare the proposed accelerated fragnet with an as-built of the same activities. The process can become complicated if the actual execution of the accelerated scenario was hampered by delays that occurred subsequent to the formulation of the acceleration scenario.

K. Specific Implementation Procedures & Enhancements

1. All Periods

The analysis is performed for each and all contemporaneous updates. Whether the periods are of fixed or variable width is dictated by the frequency of the contemporaneous updates, not by the forensic analyst.

2. Grouped Periods

The analysis is performed for grouped periods where each group may contain updates between two or more updates with the same planned critical path being compared for variance calculation. So for example, a group may be the period starting with the January update and ending with the May update, and contain three other updates (February, March, April). The three updates are not ignored but may not be directly utilized in quantifying the variance.

3. Blocked Periods

The individual periods, whether prepared in the all-periods mode or the grouped-periods mode, can be gathered into blocks for summarization. Blocking is mentioned here to distinguish the practice from grouping. Blocking is the summing of the variances obtained in several contiguous periods of an all-periods implementation, while grouping skips over the individual variance calculation for periods inside the group.

The all-periods implementation yields more information than the grouped-periods implementation and is considered more complete in that it identifies and measures the critical project delay for the entire project duration.. Also the grouped-periods implementation allows the analyst to ignore periods that may be unfavorable to the party for which the analysis is being performed by not explicitly showing the variances between the updates within each grouping.

4. Changing the Contemporaneous Project Schedule During the Analysis

MIP 3.3 is an observational technique that does not involve the insertion or deletion of delays, but instead is based on observing the behavior of the network from update to update and measuring schedule variances based on unaltered, existing logic models. The analyst’s preference is to identify and measure the critical project delays using the contemporaneous project schedules as they existed during the project.

However minor corrections to the contemporaneous schedules do not automatically result in a shift in classification of the analytical technique from MIP 3.3 to MIP 3.5 (Observational / Dynamic / Modified or Recreated). Certain limited corrections do not rise to the level of “recreations” or “modifications” and, thus, a MIP 3.3 analysis conducted using schedules with limited corrections is still properly characterized as a MIP 3.3 analysis and not a MIP 3.5 analysis. Refer to Subsection 2.3.D.3 for specific changes that can be implemented under this restriction.

The preference of every analyst should be to use the contemporaneous schedules and updates as they were prepared, reviewed, approved or accepted, and used on the project. This belief is grounded in the fact that the parties used the imperfect schedules to make decisions and manage the project work. Thus, these schedules, even though not perfect, are the best representation of the parties’ objectives and understanding of the project contemporaneously and are an indicator of each party’s performance. However, absent contract language mandating the use of the contemporaneous schedules to quantify delay, MIP 3.3 is not so rigid that corrections to the contemporaneous schedules cannot be considered by the analyst.

All corrections should be described in the analyst’s report so that the other parties and the fact finders understand the changes that the analyst made to the contemporaneous schedule.

The issue of correcting the schedule is one of balance and reasonableness and, for this reason corrections should not be made across the board or automatically. Whenever the analyst believes that changes or modifications to the contemporaneous project schedule are necessary during the analysis, it must be kept in mind that MIP 3.3, is a "self-correcting" analysis.

Finally, the analyst must also be consistent and maintain independence and objectivity. The analyst cannot limit corrections to those that have the affect of improving the analyst's client's position.

One option is to run the analysis two ways. The first run of the analysis would use the schedules as they existed contemporaneously, or unaltered. The second run of the analysis would use the schedule with the minor correction. This approach allows the finder of fact to see the difference, understand the proposed minor modification, and make a reasoned decision without having to guess what the difference would have been between the performing the analysis with the unaltered schedule and with the corrected schedule.

L. Summary of Considerations In Using the Minimum Protocol

- Cannot be implemented if contemporaneous schedule updates do not exist.

- Uses as the primary tool a set of contemporaneous schedules that are already familiar to the parties at dispute.

- Can enhance credibility if it can be shown that the project participants used the contemporaneous schedules in managing and constructing the project.

- Accounts for the dynamics of evolving events and conditions because it considers the real-time perspective of project conditions, the state of mind, and knowledge of the project participants during each update period

- Considers the dynamic nature of the critical path because it identifies shifts in the critical path between the updates

- Delays or savings in time can be assigned to specific activities.

- Data preparation process may be quicker than other methods that require a separate as-built schedule.

- This method can be used to identify and specifically quantify acceleration.

M. Caveats In Using the Minimum Protocol / Conditions Requiring Enhanced Protocols

- Actual critical path, in hindsight, may be different from that indicated as the planned critical path shown in the contemporaneous schedule updates.

- To yield accurate results, the contemporaneous schedule updates used in the analysis must be validated as accurate both in reported progress and in the network's representation of contemporaneous means and methods

- Except with very simple network models, it may be difficult to distinguish schedule variances caused by non-progress revisions from schedule variances caused purely by insufficient progress. Consider MIP 3.4 to overcome this challenge.

- If date constraints were liberally used in the update schedules, analysis may be very difficult.

3.4. Observational / Dynamic / Contemporaneous Split (MIP 3.4)

A. Description

MIP 3.4 is identical to MIP 3.3 in all respects except that for each update an intermediate file is created between the current update and the previous update consisting of progress information without any non-progress revisions. Generally, the process involves updating the previous update with progress data from the current update and recalculating the previous update using the current data date. This is the intermediate schedule or the half-step schedule. The process allows the analyst to bifurcate the update-to-update schedule variances based on pure progress by evaluating the difference between the previous update and the half-step, and then the variance based on non-progress revisions by observing the difference between the half-step and the current update.

As with MIP's 3.3, 3.4 is a retrospective technique that uses the project schedule updates to quantify the loss or gain of time along a logic path and identify the causes. Although this method is a retrospective technique, it relies on the forward-looking calculations made at the time the updates were prepared. That is, it primarily uses the information to the right of the updates' data date.

MIP 3.4 is an observational technique since it does not involve the insertion or deletion of delays, but instead is based on observing the behavior of the network from update to update and measuring schedule variances based on essentially unaltered, existing schedule logic.

Because the method uses schedule updates whose logic may have changed from the previous updates as well as from the baseline, it is considered a dynamic logic method.

It is labeled contemporaneous because the updates it relies on were prepared contemporaneously with the project execution as opposed to reconstructed after the fact as in MIP 3.5.

The 'split' label distinguishes this method from MIP 3.3 by the fact that the updates are evaluated after the bifurcation process that splits the pure progress update from the non-progress revisions.

While rare, it is possible that no non-progress revisions were made in the contemporaneous updates. If that is the case, then MIP 3.3 is a better solution for the analysis.

B. Common Names

1. Contemporaneous period analysis

2. Contemporaneous project analysis

3. Contemporaneous schedule analysis

4. Bifurcated CPA

5. Half-stepped update analysis

6. Two-stepped update analysis

7. Month-to-month

8. Window analysis

9. Windows analysis

C. Recommended Source Validation Protocols

1. Implement SVP 2.1 (baseline validation)

2. Implement SVP 2.3 (update validation)

3. Implement SVP 2.2 D.2 (as-built validation)

D. Enhanced Source Validation Protocols

1. Implement SVP 2.2 (as-built validation)
2. Implement SVP 2.4 (identification of delay events)

E. Recommended Implementation Protocols

1. Recognize all contract time extensions granted.

2. Identify the critical path activity that will be used to track the loss or gain of time for the overall network.

3. Determine whether evaluations will be done on all periods or grouped periods as described in Subsection 3.4.K.

4. Not every update needs be used, but accuracy tends to be reduced if multiple-month update periods are utilized.

5. Separately identify activities that will be used to track intra-network time losses and gains, such as on interim milestones.

6. Create a copy of the as-planned schedule and each of the update schedules for use in analysis as the bifurcated updates.

7. Import progress from the next update into each of the newly created bifurcated updates for use in identifying progress only gains and losses.

8. Compare the update at the start of the analysis period to the progress-only bifurcated update, and then compare that progress-only bifurcated update to the update at the end of the analysis period.

9. Use both the longest path and the least float criteria to identify the controlling chain of activities.

10. Identify changes (gain or lost time) in overall Project completion date, and if necessary, in interim milestone completion dates.

11. Identify start and finish variances of critical and near-critical activities in the analysis period.

12. Indentify all changes and/or revisions to logic, durations, and/or progress that were made during analysis period.

13. Sum the net gains and losses for the update at the start of the update period and the bifurcated update for that same period. The net gains and losses must equal the net gains and losses between the start of the update period and the start of the next update period.

14. Identify responsibility for delays and gains during analysis period.

15. Continue with implementation until all periods are complete

16. Sum the net gains and losses for each period to arrive at an overall impact to the project. The sum of the net impacts must be equal to difference between the first schedule update and last schedule update used in the evaluation.

F. Enhanced Implementation Protocols

1. Daily Progress Method
(See Subsection 3.3.F.1)

G. Identification of Critical & Near-Critical Paths

- Identify and understand all related contractual language.

- Identify the negative float theory being used by the opposing analyst.

- For each analysis interval, identify the calculated critical path using the longest path and the lowest total float concept of the validated update(s) corresponding to the analysis interval.

- The near-critical activity-set in each analysis interval is the one that yields the most number of activities using one of the following methods:

  - lowest float value in the update PLUS the average duration of all discrete delay events contained in whole or in part inside the analysis interval, or

  - lowest float value in the update PLUS duration of the analysis interval.

H. Identification & Quantification of Concurrent Delays & Pacing

- Determine whether compensable delay by contractor or owner is at issue.

- Identify and understand all related contractual language.

- For each delay event, distinguish the cause from the effect of delay.

- Determine whether literal or functional concurrency theory is to be used.

.

- If applicable, determine the near-critical threshold (see Subsection 4.3.)

- If applicable, determine the frequency, duration, and placement of the analysis intervals.

- For each analysis interval, identify the critical path(s) and the near-critical path(s) and explain all relative delayed starts and extended duration of activities that occurred in the previous analysis interval on the same chains of activities.

- For each suspected pacing delay event, identify the parent delay(s) and establish the order of precedence between the parent delay and the pacing delay.

- For each suspected pacing delay event, evaluate whether enough resources could have been realistically employed to perform the paced activity within its original planned duration.

I. Determination & Quantification of Excusable and Compensable Delay

(See Subsection 3.3.I)

J. Identification & Quantification of Mitigation / Constructive Acceleration

The observational / dynamic analysis methods are especially well-suited for identifying and quantifying acceleration and delay mitigation through the use of logic changes. These methods allow the analyst to not only quantify the acceleration, but also determine whether the acceleration was achieved by current, actually implemented measures, or by logic changes representing the promise of future acceleration.

The difference between this method and MIP 3.3 is that the bifurcation of each update into half-steps in MIP 3.4 makes it much easier to identify acceleration and delay mitigation that results from logic changes.

As with MIP 3.3, in 3.4, acceleration or delay mitigation is identified by comparing the completion date of the longest path of the previous period with that of the current period. A current date that is earlier than the previous date suggests acceleration. However, note that the value is a net number potentially representing both slippage and gain, where the gain was greater than the slippage. Thus, a detailed examination of the longest path, the near-longest path surrounding the data date, and the examination of the logic changes between the last and the current periods along those paths are necessary for a competent identification and quantification of acceleration and delay mitigation.

In order to determine whether the promised future acceleration was actually implemented, it will be necessary to compare the proposed accelerated fragnet with an as-built of the same activities. The process can become complicated if the actual execution of the accelerated scenario was hampered by delays that occurred subsequent to the formulation of the acceleration scenario.

K. Specific Implementation Procedures & Enhancements

1. All Periods

The analysis is performed for each and all contemporaneous updates. Whether the periods are of fixed or variable width is dictated by the frequency of the contemporaneous updates, not by the forensic analyst.

2. Grouped Periods

The analysis is performed for grouped periods where each group may contain updates between the two updates being compared for variance calculation. So for example, a group may be the period starting with the January update and ending with the May update, and contain three other updates (February, March, April). The three updates are analyzed just as they would be analyzed if they were not grouped and the results would be the same, whether grouped or not.

3. Blocked Periods

The individual periods, whether prepared in the all-periods mode or the grouped-periods mode can be gathered into blocks for summarization. Blocking is mentioned here to distinguish the practice from grouping.

The all-periods implementation yields more information than the grouped-periods implementation and is considered more complete in that it identifies and measures the critical project delay for the entire project duration.. Also the grouped-periods implementation allows the analyst to ignore periods that may be unfavorable to the party for which the analysis is being performed by not explicitly showing the variances between the updates within each grouping.

4. Bifurcation: Creating a Progress-Only Half-Step Update

Bifurcation (a.k.a. half-stepping or two-stepping) is a procedure to segregate progress reporting from various non-progress revisions inherent in the updating process. Elements that are considered to be non-progress revisions include:

- Addition or deletion of activities
- Split or combined activities, using new activity IDs
- Addition or deletion of logic links
- Changes to lag value of logic links
- Addition, deletion or changes to constraints
- Changes to OD
- Increase in RD such that RD becomes greater than OD
- Changes to RD not accompanied by changes to PCT
- Increase in RD of activities that have not started
- Changes to calendar assignments
- Changes to holiday assignments within a pre-existing calendar

The following is one of several step-by-step procedures used to perform the bifurcation:

a. Make a copy of the baseline or an updated schedule for which a half-step is to be created. The original baseline or update will be referred to herein as 01 and the copy as H1.

b. Update the copy, H1, using the progress data from the next schedule update [referred to herein as 02] for the following fields:

   i. Actual start

   ii. Actual finish

   iii. Increased percent complete

   iv. Decreased remaining duration

c. Recalculate schedule H1 by setting the data date[5] to that used by 02.

d. The variance between the completion dates of H1 compared to that of 01 represents the slippage or gain due to progress during the update period.

e. The variance between the completion dates of H1 compared to that of 02 represents the slippage or gain due to non-progress revisions made in 02.

f. These two variance values add up to the variance between 01 and 02.

g. The validity of the H1 file should be checked by comparing the duration of the update period (that is, the difference between the two data dates) to the progress variance. If the progress variance value is greater than the duration of the update period, there are two possible explanations:

   i. The first one is that there is a 'pseudo-non-progress revision' such as an increase in RD-value found itself in the H1 file. This needs to be fixed.

   ii. The second possibility is that the lack of progress during the update period pushed subsequent activities into a period of no-work defined by the calendar. This does not need to be fixed.

h. Elements that are considered to be nuisances or complications that require case-by-case intervention by the analyst include:

   i. Significant changes in activity descriptions to a schedule activity occupying a preexisting activity ID

   ii. Assignments of a different activity ID to a preexisting schedule activity

   iii. Changes in actual start or actual finish values previously reported

   iv. Any change in calculation mode such as progress override and retained logic

Reversal of previously reported progress (i.e. deprogressing) by either increasing the value of remaining duration of the activity over the previously stated value or decreasing the percentage-complete value under what was previously reporte**d.**

[5] Note that in some software packages, for example, Microsoft Project, the default setting need to be changed to recognize the concept of the data date.

5. Changing the Contemporaneous Project Schedule During the Analysis

MIP 3.4 is an observational technique that does not involve the insertion or deletion of delays, but instead is based on observing the behavior of the network from update to update and measuring schedule variances based on unaltered, existing logic models. The analyst's preference is to identify and measure the critical project delays using the contemporaneous project schedules as they existed during the project.

However minor corrections to the contemporaneous schedules do not automatically result in a shift in classification of the analytical technique from MIP 3.4 to MIP 3.5 (Observational / Dynamic / Modified or Recreated). Certain limited corrections do not rise to the level of "recreations" or "modifications" and, thus, a MIP 3.4 analysis conducted using schedules with limited corrections is still properly characterized as a MIP 3.4 analysis and not a MIP 3.5 analysis. Refer to Subsection 2.3.D.3 for specific changes that can be implemented under this restriction.

The preference of every analyst should be to use the contemporaneous schedules and updates as they were prepared, reviewed, approved or accepted, and used on the project. This belief is grounded in the fact that the parties used the imperfect schedules to make decisions and manage the project work. Thus, these schedules, even though not perfect, are the best representation of the parties' objectives and understanding of the project contemporaneously and are an indicator of each party's performance. However, absent contract language mandating the use of the contemporaneous schedules to quantify delay, MIP 3.4 is not so rigid that corrections to the contemporaneous schedules cannot be considered by the analyst.

All corrections should be described in the analyst's report so that the other parties and the fact finders understand the changes that the analyst made to the contemporaneous schedule.

The issue of correcting the schedule is one of balance and reasonableness and, for this reason corrections should not be made across the board or automatically. Whenever the analyst believes that changes or modifications need to be implemented in the contemporaneous project schedules during the analysis, it should be noted that MIP 3.3, is a "self-correcting" analysis since it uses each of the successive contemporaneous schedule updates rather than progressing a single schedule.

Finally, the analyst must also be consistent and maintain independence and objectivity. The analyst cannot limit corrections to those that have the affect of improving the analyst's client's position.

One option is to run the analysis two ways. The first run of the analysis would use the schedules as they existed contemporaneously, or unaltered. The second run of the analysis would use the schedule with the minor correction. This approach allows the finder of fact to see the difference, understand the proposed minor modification, and make a reasoned decision without having to guess what the difference would have been between the performing the analysis with the unaltered schedule and with the corrected schedule.

L. Summary of Considerations In Using the Minimum Protocol

- Allows for easier identification of schedule slippage and gains due to schedule revisions and other non-progress factors compared to MIP 3.3

- Cannot be implemented if contemporaneous schedule updates do not exist.

- Uses as the primary tool a set of contemporaneous schedules that are already familiar to the parties at dispute.

- Can enhance credibility if it can be shown that the project participants used the contemporaneous schedules in managing and constructing the project.

- Accounts for the dynamics of evolving events and conditions because it considers the real-time perspective of project conditions, the state of mind, and knowledge of the project participants during each update period

- Considers the dynamic nature of the critical path because it identifies shifts in the critical path between the updates

- Delays or savings in time can be assigned to specific activities.

- Data preparation process may be quicker than other methods that require a separate as-built schedule.

- This method can be used to identify and specifically quantify acceleration.

M. Caveats In Using the Minimum Protocol / Conditions Requiring Enhanced Protocols

- Actual critical path, in hindsight, may be different from that indicated as the planned critical path shown in the contemporaneous schedule updates.

- To yield accurate results, the contemporaneous schedule updates used in the analysis must be validated as accurate both in reported progress and in the network's representation of contemporaneous means and methods

- If date constraints were liberally used in the update schedules, analysis may be very difficult.

3.5. Observational / Dynamic / Modified or Recreated (MIP 3.5)

A. Description

MIP 3.5 looks like MIPs 3.3 or 3.4 except that it uses contemporaneous schedule updates that were extensively modified or 'updates' that were completely recreated. MIP 3.5 is usually implemented when contemporaneous updates are not available or never existed. The fact that it does not use the contemporaneous updates places this method in a fundamentally different category from the standpoint of the nature of source input data.

It is a retrospective technique that uses the modified or recreated schedule updates to quantify the loss or gain of time along a logic path and identify the causes. Although this method is a retrospective technique, it relies on the forward-looking calculations made at the time the updates would have been prepared. That is, it primarily uses the information to the right of the updates' data date.

While MIP 3.5 is still categorized as an observational technique since it does not involve the insertion or deletion of delays, it is not purely observational when seen in the context of the level of data intervention by the analyst. MIP's 3.3 and 3.4 are purely observational in the sense that the analyst is interpreting what is observed in the behavior of the network from update to update and measuring schedule variances based on unaltered, existing logic models. Because of extensive data intervention by the analyst when using MIP 3.5, the observation is made on the behavior of the networks on which the analyst had significant control.

If there were non-progress revisions to the baseline during the project, the method must recognize those non-progress revisions. Otherwise, the modification or the reconstruction is not complete or proper. As such, a properly implemented MIP 3.5 is considered a Dynamic Logic method. If non-progress revisions did not occur on the project, the results of MIP 3.5 would be very similar to one that would result from MIP 3.2.

MIP 3.5 can be implemented with or without the half-step process. Unlike the contemporaneous MIP's 3.3 and 3.4, the label 'as-is' is an irrelevant distinction from the 'split.' This is because the modification or reconstruction is under the control of the analyst.

B. Common Names

1. Update analysis

2. Reconstructed update analysis

3. Modified update analysis

4. Month-to-month

5. Window analysis

6. Windows analysis

C. Recommended Source Validation Protocols

1. Implement SVP 2.3 (update validation) and,

2. Implement SVP 2.3 D.1 or D.2 (reconstruction) and,

3. Implement SVP 2.1 (baseline validation).

D. Enhanced Source Validation Protocols

1. Implement SVP 2.2 (as-built validation)
2. Implement SVP 2.4 (identification of delay events)

E. Recommended Implementation Protocols

1. Recognize all contract time extensions granted.

2. Identify the critical path activity that will be used to track the loss or gain of time for the overall network.

3. Determine whether evaluations will be done on all periods or grouped periods as described in Subsection 3.3.K.

4. Not every update needs be used, but accuracy tends to be reduced if multiple-month update periods are utilized.

5. Separately identify activities that will be used to track intra-network time losses and gains, such as on interim milestones.
6. Compare the update at the start of the analysis period to the update at the end of the analysis period.

7. Use both the longest path and the least float criteria to identify the controlling chain of activities.

8. Identify changes (gain or lost time) in overall Project completion date, and if necessary, in interim milestone completion dates.
9. Identify start and finish variances of critical and near-critical activities in the analysis period.
10. Indentify all changes and/or revisions to logic, durations, and/or progress that were made during analysis period.
11. Identify responsibility for delays and gains during analysis period.
12. Continue with implementation until all periods are complete
13. Sum the net gains and losses for each period to arrive at an overall impact to the project. The sum of the net impacts must be equal to difference between the first schedule update and last schedule update used in the evaluation.

F. Enhanced Implementation Protocols

1. Daily Progress Method

(See Subsection 3.3.F.1)

G. Identification of Critical & Near-Critical Paths

- Identify and understand all related contractual language.
- Identify the negative float theory being used by the opposing analyst.
- For each analysis interval, identify the calculated critical path using the longest path and the lowest total float concept of the validated update(s) corresponding to the analysis interval.
- The near-critical activity-set in each analysis interval is the one that yields the most number of activities using one of the following methods:
  - lowest float value in the update PLUS the average duration of all discrete delay events contained in whole or in part inside the analysis interval, or
  - lowest float value in the update PLUS duration of the analysis interval.

H. Identification & Quantification of Concurrent Delays & Pacing

- Determine whether compensable delay by contractor or owner is at issue.
- Identify and understand all related contractual language.
- For each delay event, distinguish the cause from the effect of delay.
- Determine whether literal or functional concurrency theory is to be used.
- If applicable, determine the near-critical threshold (see Subsection 4.3).
- If applicable, determine the frequency, duration, and placement of the analysis intervals.
- For each analysis interval, identify the critical path(s) and the near-critical path(s) and explain all relative delayed starts and extended duration of activities that occurred in the previous analysis interval on the same chains of activities.
- In cases where the difference in full-hindsight approach versus 'blindsight' approach results in a significance variance, use both approaches for evaluation of concurrency.
- For each suspected pacing delay event, identify the parent delay(s) and establish the order of precedence between the parent delay and the pacing delay.
- For each suspected pacing delay event, evaluate whether enough resources could have been realistically employed to perform the paced activity within its original planned duration.

I. Determination & Quantification of Excusable and Compensable Delay

(See MIP 3.3.)

J. Identification & Quantification of Mitigation / Constructive Acceleration

(See MIP 3.3.)

K. Specific Implementation Procedures & Enhancements

1. Fixed Periods

The analysis periods are of virtually identical duration and may coincide with regular schedule update periods. Note that the fixed period implementation can be further processed into Grouped or Blocked implementation as described in MIP's 3.3 and 3.4.

2. Variable Periods

The analysis periods are of varying durations and are characterized by their different natures such as the type of work being performed, the types of delaying influences, or the operative contractual schedule under which the work was being performed.

3. Fixed-Periods vs. Variable-Periods

Similar to the comparison between the all-periods implementation and the grouped-periods implementation for MIP's 3.3 and 3.4, a frequent-fixed-periods implementation yields more information than the infrequent-variable-periods implementation, and is considered more precise. The infrequent-variable-periods implementation allows the analyst to skip over periods that may be unfavorable to the party for which the analysis is being performed.

L. Summary of Considerations In Using the Minimum Protocol

- Able to simulate MIP's 3.3 and/or 3.4 without the benefit of reliable contemporaneous schedule updates if update modification and/or reconstruction is reliable.
- Requires, at the least, a baseline schedule and a reliable source of as-built dates.
- Typically, the smaller the number of modifications to the contemporaneous schedule updates, the more credible the results of the analysis.
- Allows for the consideration of the dynamic nature of the critical path because it identifies shifts in the critical path between the updates even if reliable contemporaneous schedule updates do not exist.
- Allows for the use of hindsight progress updates to simulate the actual critical path.
- Delays can be assigned to specific activities.
- Data preparation process may be quicker than other methods that require compilation of a separate detailed as-built schedule.
- This method can be used to identify acceleration.

M. Caveats In Using the Minimum Protocol / Conditions Requiring Enhanced Protocols

- Where updates are recreated, it is perceived to be an after-the-fact analysis that fails to consider logic changes that would have been incorporated in view of contemporaneous project circumstances.
- Does not have the benefit of source schedules that are already familiar to the parties at dispute.

- To be credible, recreated schedule updates must be accurate both in reported progress to date and in the network's representation of contemporaneous means, and consistent with other project documentation during the update periods reflecting the real-time perspective of project conditions, the state of mind, and knowledge of the project participants

- Progress reported for activity performance spanning more than one period must be supported by reasonable means.

- Relatively time consuming and therefore costly to implement compared to MIP's 3.3 or 3.4 because it requires substantial support to justify the modifications or the reconstruction.

- The analyst should anticipate significantly more scrutiny and challenges regarding the reliability of the data and logic.

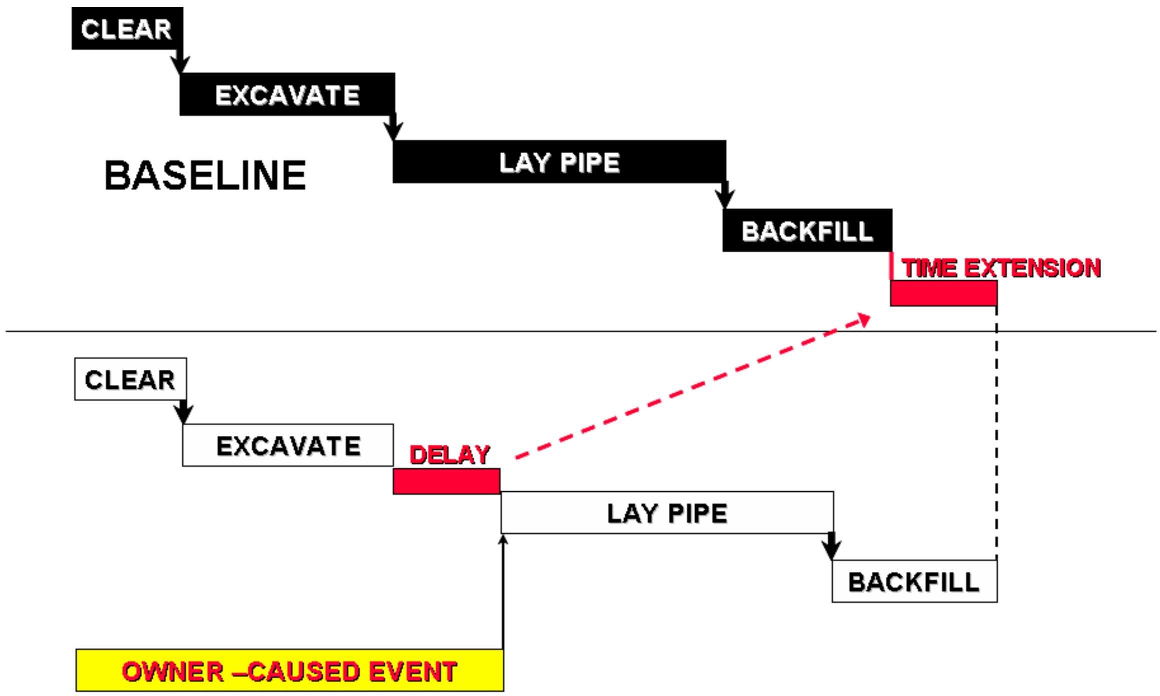
- Actual critical path, in hindsight, may be different from that indicated as the planned critical path shown in the contemporaneous schedule updates.

- Except with very simple network models, it may be difficult to distinguish schedule variances caused by non-progress revisions from schedule variances caused purely by insufficient progress.

3.6. Modeled / Additive / Single Base (MIP 3.6)

A. Description

MIP 3.6 is a modeled technique since it relies on a simulation of a scenario based on a CPM model. The simulation consists of the insertion or addition of activities representing delays or changes into a network analysis model representing a plan to determine the impact of those inserted activities to the network. Hence, it is an additive model.



Delaying event inserted into the baseline.

Figure 5 – Graphic Example: Modeled, Additive, Single Base

MIP 3.6 is a single base method, distinguished from MIP 3.7 as a multiple base method. The additive simulation is performed on one network analysis model representing the plan. Hence, it is a static logic method as opposed to a dynamic logic method.

MIP 3.6 can be used prospectively or retrospectively. Prospectively, it can be used to forecast future impacts; for description and implementation[6], see AACE Recommended Practice 52R-06, Time Impact Analysis – As Applied in Construction. Retrospectively, as described here, it relies on the forward-looking calculations to the right of the data date.

B. Common Names

1. Impacted as-planned (IAP)

2. Impacted baseline (IB)

3. Plan plus delay

6. See AACE Recommended Practice 52R-06, Time Impact Analysis – As Applied in Construction.

4. Impacted update analysis

5. Time impact analysis (TIA)

6. Time impact evaluation (TIE)

7. Fragnet insertion

8. Fragnet analysis

C. Recommended Source Validation Protocols

1. Implement SVP 2.1 (baseline validation) or,

2. Implement SVP 2.3 (update validation) and,

3. Implement SVP 2.4 (delay ID and quantification).

D. Enhanced Source Validation Protocols

1. Implement SVP 2.2 (as-built validation)

E. Recommended Implementation Protocols

1. Recognize all contract time extensions granted.

2. Identify and quantify delays that are to be evaluated, including source documents on which they are based.

3. Select the planned network to be utilized as the "un-impacted schedule". If not using the baseline, select the contemporaneous update that existed just prior to the initial delay that is to be evaluated.

4. Insert an activity or activities (fragnet) into the "un-impacted schedule" to represent the selected delay(s).

5. Calculate or schedule the new schedule created by the "un-impacted schedule" with the fragnet or activity inserted. In the most basic implementations (i.e. bar chart evaluation) it may be necessary to calculate the impact by hand. The resultant network is considered the "impacted schedule".

6. Zero out the durations of all activities in the added fragnet and verify that when calculated, there is no change to the completion date from the un-impacted schedule completion date. This verifies that there is no added logic in the fragnet that creates a delay.

7. Ensure that the resulting schedule has at least one continuous critical path, using the longest path criterion that starts at NTP or some earlier start milestone and ends at a finish milestone, which is the latest occurring schedule activity in the network, after the insertion of delay activities.

8. Compare the Project completion date of the impacted and un-impacted schedules to determine the impact of the inserted fragnet(s).

9. Tabulate and justify each change made to the baseline used to create the impacted as-planned.

10. Use both the longest path and the least float criteria to identify the controlling chain of activities.

11. Quantify net delays and gains.

F. Enhanced Implementation Protocols

1. Analysis accompanied by a listing of known significant delays that are not incorporated into the model.

2. Compare the impacted schedule to the as-built and explain the variances between the two schedules for all significant chains of activities.

G. Identification of Critical & Near-Critical Paths

- Identify and understand all related contractual language.

- Identify the negative float theory being used by the opposing analyst.

- From the baseline schedule, identify the calculated critical path of the baseline using the longest path and the lowest total float concept of the validated baseline.

- The near-critical activity-set is the one that yields the most number of activities using one of the following methods:

  - the lowest float value in the pre-insertion baseline network PLUS the maximum duration of all the inserted delays, or

  - the float value of the pre-insertion baseline longest path PLUS the maximum duration of all the inserted delays, or

  - the lowest float value in the pre-insertion baseline PLUS the average duration of the periods of schedule updates or revisions generated during the project.

- Stepped insertion should be in chronological order of the occurrence of the delay event.

H. Identification & Quantification of Concurrent Delays & Pacing

In its minimum implementation, concurrency cannot be evaluated by this method. The procedure below outlines some enhancements over the minimum implementation that would allow limited evaluation of concurrent delays using this method.

- Determine whether compensable delay by contractor or owner is at issue.

- Identify and understand all related contractual language.

- For each delay event, distinguish the cause from the effect of delay.

- Determine whether literal or functional concurrency theory is to be used.

- If applicable, determine the near-critical threshold (see Subsection 4.3).

- If applicable, determine the frequency, duration, and placement of the analysis intervals.

- Compare the pre-insertion baseline to the as-built and discretely identify and classify by causation all delays on those chains of activities that are near-critical in the pre-insertion baseline schedule.

- Insert the delays found in the previous step into the pre-insertion baseline and compare the result with the impacted baseline that resulted from the insertion of the claimed delays.

- For each suspected pacing delay event, identify the parent delay(s) and establish the order of precedence between the parent delay and the pacing delay.

- For each suspected pacing delay event, evaluate whether enough resources could have been realistically employed to perform the paced activity within its original planned duration.

I. Determination & Quantification of Excusable and Compensable Delay

Identify and understand all contractual language related to delay apportionment and determine whether the contractual language would override any determination of excusability and compensability based on findings resulting from analyses performed under this RP.

Absent contract language or other agreements, use the following procedure to determine the net total delay apportionment:

1. Excusable & Compensable Delay (ECD)

An additive-modeled schedule by itself does not account for concurrent delays and is therefore unsuitable for determining compensability to the claimant. However, it is possible to analyze for *approximate* concurrency by comparing two additive-modeled schedules. To do this:

a. Create one additive model by inserting all owner-caused and *force majeure*-caused impact events into the baseline.

b. Create another additive model by inserting all contractor-caused impact events into the baseline.

c. Compare the two resulting schedules. To the extent that the net delay-effect beyond the baseline completion date overlaps, there is concurrency.

d. The extent to which the completion date of the additive model with the owner-impact is later than that of the other additive model with the contractor-impact, *may* be the quantity of ECD, but only to the extent that the impacted completion date does not exceed the actual completion date.

2. Non-Excusable & Non-Compensable Delay (NND)

An additive-modeled schedule by itself does not account for concurrent delays and is therefore unsuitable for determining compensability to the respondent or liquidated/stipulated damages. However, it is possible to analyze for *approximate* concurrency by comparing two additive-modeled schedules. To do this:

a. Create one additive model by inserting all owner-caused and *force majeure*-caused impact events into the baseline.

b. Create another additive model by inserting all contractor-caused impact events into the baseline.

c. Compare the two resulting schedules. To the extent that the net delay-effect beyond the baseline completion date overlaps there is concurrency.

d. The extent to which the completion date of the additive model with the contractor-impact is later than that of the other additive model with the owner-impact, *may* be the quantity of NND, but only to the extent that the impacted completion date does not exceed the actual completion date.

3. Excusable & Non-Compensable Delay (END)

a. Insert all owner-caused and *force majeure*-caused impact events into the baseline and recalculate the schedule.

b. The difference between the baseline completion of the longest path and the completion of the longest path in the additive model is the END.

c. If the completion of the longest path in the additive model is later than the actual completion date, the END is the difference between the baseline completion and the actual completion dates.

J. Identification & Quantification of Mitigation / Constructive Acceleration

The comparison between the completion date of the longest path of the additive model and the actual completion date will provide a gross approximation of acceleration or delay mitigation. This is based on the theory that if non-contractor delays inserted into the baseline yield a completion date that is later than that actually achieved, it must have resulted from shortening of actual performance duration and/or the use of more aggressive logic. Note that the gross comparison does not provide the detail necessary in order to address the issue of who gets the credit for the acceleration.

K. Specific Implementation Procedures & Enhancements

1. Global Insertion

Once the Baseline Schedule is identified then all known delaying events are added to this schedule. In the global insertion method, all delay events and influences are added together and the impact is determined on the combined effect of the added delays. If the analyst is trying to document the total impact of all delay events then insertion of all events at one time may accomplish this task.

2. Stepped Insertion

The delays are added individually or in groups to the Baseline Schedule and the impact is determined after each iterative insertion. If the analyst is concerned with the impact of each delay event then the events should be inserted in chronological order of occurrence of the event in order to reflect actual circumstances If the events are introduced into the delay analysis individually, the impacted completion date should be recorded after each delay is included.

For each delay event introduced into this analysis one must be able to document the duration of the delay, and the predecessor and successor activities related to the delay, in order to perform this method objectively.

L. Summary of Considerations In Using the Minimum Protocol

- Suited primarily for the use in identifying and quantifying potential delays rather than actual delays.
- This method can be used to quantify non-compensable time extensions, but cannot, by itself, quantify compensable delays because it does not account for concurrent or pacing delays.
- This method can be used to identify acceleration, although actual performance that is better than predicted by use of this method does not, in and of itself, necessarily demonstrate active implementation of acceleratory measures.
- Intuitively easy to understand and present, and can be understood especially by those that do not have a construction background.
- Does not require an as-built schedule or contemporaneous schedule updates.
- Can be implemented relatively easily and quickly compared to other MIP's, but is of limited reliable use.

M. Caveats In Using the Minimum Protocol / Conditions Requiring Enhanced Protocols

- Because it does not rely on as-built data, it is a hypothetical model, especially where the project is actually constructed differently than the baseline schedule logic.
- Susceptible to unintended or intended manipulation due to modeling if only one party's delays are considered, since the method cannot account for the impact of delays not explicitly inserted.
- Accuracy of the duration of critical path impact for any given delay event degrades in proportion to the chronological distance of the delay event from the data date of the schedule.
- Since it relies only on the initial as-planned critical path to analyze delays, it does not account for changes in logic or durations of activities
- Does not necessarily consider the chronological order of delays.
- Extremely sensitive to the order of fragnet and logic insertion.

3.7. Modeled / Additive / Multiple Base (MIP 3.7)

A. Description

MIP 3.7 is a modeled technique since it relies on a simulation of a scenario based on a CPM model. The simulation consists of the insertion or addition of activities representing delays or changes into a network analysis model representing a plan to determine the impact of those inserted activities to the network. Hence, it is an additive model.

MIP 3.7 is a multiple base method, distinguished from MIP 3.6 as a single base method. The additive simulation is performed on multiple network analysis models representing the plan, typically an update schedule, contemporaneous, modified contemporaneous, or recreated. Each base model creates a period of analysis that confines the quantification of delay impact.

Because the updates typically reflect non-progress revisions, it is a dynamic logic method as opposed to a static logic method.

MIP 3.7 is a retrospective analysis since the existence of the multiple periods means the analyst has the benefit of hindsight.

B. Common Names

1. Window analysis

2. Windows analysis

3. Impacted update analysis

4. Time impact analysis (TIA)

5. Time impact evaluation (TIE)

6. Fragnet insertion

7. Fragnet analysis

C. Recommended Source Validation Protocols

1. Implement SVP 2.1 (baseline validation) and,

2. Implement SVP 2.3 (update validation) and,

3. Implement SVP 2.4 (delay ID and quantification)

D. Enhanced Source Validation Protocols

1. Implement SVP 2.2 (as-built validation)

E. Recommended Implementation Protocols

1. Recognize all contract time extensions granted.

2. Identify and quantify delays that are to be evaluated, including source documents on which they are based.

3. Select the as-planned network to be utilized as the “un-impacted schedule”. If not using the baseline, select the contemporaneous update that existed just prior to the initial delay that is to be evaluated.

4. Identify the schedule updates, or recreated updates, that correlate to the beginning of each analysis interval.

5. Insert an activity or activities (fragnet) into the “un-impacted schedule” to represent the selected delay(s). Ensure that the impact events are chronologically inserted into the proper update schedules.

6. Calculate or schedule the new schedule created by the “un-impacted schedule” with the fragnet or activity inserted. In the most basic implementations (i.e. bar chart evaluation) it may be necessary to calculate the impact by hand. The resultant network is considered the “impacted schedule”.

7. Zero out the durations of all activities in the added fragnet and verify that when calculated, there is no change to the completion date from the un-impacted schedule completion date. This verifies that there is no added logic in the fragnet that creates a delay situation.

8. Ensure that the resulting schedule has at least one continuous critical path, using the longest path criterion that starts at NTP or some earlier start milestone and ends at a finish milestone, which is the latest occurring schedule activity in the network, after the insertion of delay activities.

9. Tabulate and justify each change made to an update schedule to create the impacted schedule. Insert model fragnets in the correct updated schedule containing previous impacts, period by period.

10. Use both the longest path and the least float criteria to identify the controlling chain of activities.

.

11. A new analysis period needs to be established with each significant change in the critical path chain of activities, and with each available contemporaneous update schedule

12. Correlate the impacted schedule with each available contemporaneous update, identifying and using either hindsight or blindsight for establishing remaining durations for the incomplete fragnet activities.

13. Quantify net delays and gains.

14. Prepare a tabulation that summarizes the variances quantified for each analysis period and reconcile the total to the result that would be obtained by a competent implementation of MIP 3.1.

F. Enhanced Implementation Protocols

1. Analysis is accompanied by a listing of known significant delays not incorporated into the model.

2. Compare the impacted schedule to the as-built and explain the variances between the two schedules for all significant chains of activities.

3. Use accepted baseline, updates and schedule revisions.

G. Identification of Critical & Near-Critical Paths

- Identify and understand all related contractual language.

- Identify the negative float theory being used by the opposing analyst.

- For each analysis interval, identify the calculated critical path using the longest path and the lowest total float concept of the <u>pre-insertion</u> validated update(s) corresponding to the analysis interval.

- The near-critical activity-set in each analysis interval is the one that yields the most number of activities using one of the following methods:

  - float value of the longest path in the <u>pre-insertion</u> validated update PLUS the maximum duration of all discrete delay events inserted in whole or in part inside the analysis interval, or

  - lowest float value in the <u>pre-insertion</u> validated update PLUS the maximum duration of all discrete delay events inserted in whole or in part inside the analysis interval, or

  - lowest float value in the update PLUS duration of the analysis interval.

- Stepped insertion should be in chronological order of the occurrence of the delay event.

H. Identification & Quantification of Concurrent Delays & Pacing

- Determine whether compensable delay by contractor or owner is at issue.

- Identify and understand all related contractual language.

- For each delay event, distinguish the cause from the effect of delay.

- Determine whether literal or functional concurrency theory is to be used.

- If applicable, determine the near-critical threshold (see Subsection 4.3.)

- If applicable, determine the frequency, duration, and placement of the analysis intervals.

- For each analysis interval, compare the pre-insertion schedule update(s) corresponding to the analysis interval to the as-built, and discretely identify and classify by causation all delays on those chains of activities that are near-critical in the pre-insertion schedule update.

- Insert those discrete delay activities into the pre-insertion update and compare the result of the impacted schedule to the un-impacted schedule for that analysis interval that resulted from the insertion of the claimed delays.

- Compare the longest path of the impacted schedule for the analysis interval with the longest path of the same schedule recalculated with the progress data and the data date of the subsequent analysis interval. If the longest path and the overall completion dates are the same, the predictive model generated for the analysis period is reasonably accurate.

.

- If the longest path is the same but the overall completion date of the progressed version is later, the delay predicted for the longest path was, in actuality, worse, or additional delay events occurred on the longest path.

- If the longest path is the same but the overall completion date of the progressed version is earlier, there was acceleration or some other delay mitigation on the delays on the longest path.

- If the longest path and the overall completion dates are the same but an additional path is also the longest path, some activity or delay event on that additional longest path may be concurrent with the claimed delay.

- If the longest path has changed but the overall completion date is the same, some activity or delay event on the new longest path may be partially or completely concurrent with the claimed delay on the former longest path.

- If the longest path has changed but the overall completion date is earlier, some activity or delay event on that new longest path may be partially or completely concurrent with the claimed delay on the former longest path.

- If the longest path has changed but the overall completion date is later, some activity or delay event on that new longest path may be partially or completely concurrent with the claimed delay on the former longest path.

- Compare the longest path of the progressed version of the analysis interval with the longest path of the pre-insertion baseline of the subsequent analysis interval. Any differences are the result of non-progress revisions implemented in the pre-insertion baseline of the subsequent analysis interval and should be identified and explained.

- Repeat the process for all analysis intervals.

- For each suspected pacing delay event, identify the parent delay(s) and establish the order of precedence between the parent delay and the pacing delay.

- For each suspected pacing delay event, evaluate whether enough resources could have been realistically employed to perform the paced activity within its original planned duration.

I. Determination & Quantification of Excusable and Compensable Delay

Identify and understand all contractual language related to delay apportionment and determine whether the contractual language would override any determination of excusability and compensability based on findings resulting from analyses performed under this RP. Note that this method or a variation of this is often specified as the method of choice in many construction contracts, including specific procedural steps for implementation. Therefore, the following procedure should be applied only in absence of contract language or other agreements.

1. Excusable & Compensable Delay (ECD)

An additive-modeled schedule by itself does not account for concurrent delays and is therefore unsuitable for determining compensability. However, it is possible to analyze for concurrency by comparing two additive-modeled schedules. The reliability of this quantification method is inversely proportional to the duration of the analysis periods. In other words, the shorter the period duration, the more reliable the quantification. See Subsection 4.2.D.4.

To do this, for each analysis period:

a. Create one additive model by inserting the subject owner-caused and *force majeure*-caused impact events into the update with the data date closest in time prior to the commencement of the impact event.

Create a separate additive model by inserting the contractor-caused impact events into the same update chosen for the owner-impact model.

b. Compare the two resulting schedules. To the extent that the net delay-effect beyond the baseline completion date overlaps there is concurrency.

c. The extent to which the completion date of the additive model with the owner-impact is later than that of the other additive model with the contractor-impact, *may* be the quantity of ECD, but only to the extent that the impacted completion date does not exceed the actual completion date.

2. Non-Excusable & Non-Compensable Delay (NND)

An additive-modeled schedule by itself does not account for concurrent delays and is therefore unsuitable for determining compensability. However, it is possible to analyze for concurrency by comparing two additive-modeled schedules. The reliability of this quantification method is inversely proportional to the duration of the analysis periods. In other words, the shorter the period duration, the more reliable the quantification. See Subsection 4.2.D.4.

To do this, for each analysis period:

a. Create one additive model by inserting the subject contractor-caused impact events into the update with the data date closest in time prior to the commencement of the impact event.

b. Create a separate additive model by inserting the owner-caused and *force majeure*-caused impact events into the same update chosen for the owner-impact model.

c. Compare the two resulting schedules. To the extent that the net delay-effect beyond the baseline completion date overlaps there is concurrency.

d. The extent to which the completion date of the additive model with the contractor-impact is later than that of the other additive model with the owner-impact, *may* be the quantity of NND, but only to the extent that the impacted completion date does not exceed the actual completion date.

3. Excusable & Non-Compensable Delay (END)

a. Insert the owner-caused and *force majeure*-caused impact events into the update with the data date closest in time prior to the commencement of the impact event.

b. The difference between the completion of the longest path prior to the insertion and the completion of the longest path after the insertion is the END.

c. The post-insertion schedule can be further analyzed by inserting actual progress data. If the resulting completion date is shorter than that indicated in the post-insertion schedule prior to actual progressing, it may be proper to reduce the amount of END accordingly.

J. Identification & Quantification of Mitigation / Constructive Acceleration

In MIP 3.7, after inserting delays into the update closest in time preceding the delay, the identity and the movement of the critical path is monitored. Then, when the update is progressed with actual progress data and the same logic path reexamined, if the logic path is shorter than that which was calculated prior to adding actual progress, there was acceleration or schedule recovery during the period for which actual progress was entered.

K. Specific Implementation Procedures & Enhancements

1. Fixed Periods

The analysis periods are of virtually identical duration and may coincide with regular schedule update periods.

2. Variable Periods

The analysis periods are of varying durations and are characterized by their different natures such as the type of work being performed, the types of delaying influences, significant project events, changes to the critical path, revised baseline schedules, and/or the operative contractual schedule under which the work was being performed.

3. Global Insertion

All the delay events and influences are added together and the impact is determined on the combined effect of the added delays.

4. Stepped Insertion

The delays are added individually or in groups and the impact is determined after each iterative insertion. Note that stepping is different from inserting the delays in time period groups that create a straight, vertical delineation of analysis periods; whereas, delays for each step insertion may not fit neatly into an existing analysis period.

L. Summary of Considerations In Using the Minimum Protocol

- Considers the chronological order of delays better than MIP 3.6.
- Can be performed relatively easily throughout the life of the project for project control when implemented as the AACE Recommended Practice 52R-06, Time Impact Analysis, as well as for forensic use as described in this recommended practice.
- Takes into consideration changes to the critical path as they occur on the project
- Requires routine schedule updates performed throughout project life.

- This method can be used to quantify non-compensable time extensions, but cannot, by itself, quantify compensable delays because it does not account for concurrent or pacing delays.

- This method can be used to identify and quantify acceleration, although actual performance that is better than predicted by use of this method does not, in and of itself, necessarily demonstrate active implementation of acceleratory measures.

- Does not require an as-built schedule.

M. Caveats In Using the Minimum Protocol / Conditions Requiring Enhanced Protocols

- Because it does not rely on as-built data, it is a hypothetical model, especially where the project is actually constructed differently than the baseline schedule logic. However, compared to MIP 3.6, the periodic nature of the analysis incorporates as-built data.

- Susceptible to unintended or intended manipulation due to modeling if only one party's delays are considered, since the method cannot account for the impact of delays not explicitly inserted.

- Accuracy of the duration of critical path impact for any given delay event degrades in proportion to the chronological distance of the delay event from the data date of the schedule.

- Labor intensive in comparison to MIP 3.6 when implemented properly because of the additional source schedules and technical complexity.

- Extremely sensitive to the order of fragnet and logic insertion.

3.8. Modeled / Subtractive / Single Simulation (MIP 3.8)

A. Description

3.8 is a modeled technique relying on a simulation of a scenario based on a CPM model. The simulation consists of the extraction of entire activities or a portion of the as-built durations representing delays or changes from a network analysis model representing the as-built condition of the schedule to determine the impact of those extracted activities on the network. Hence, it is a subtractive model.

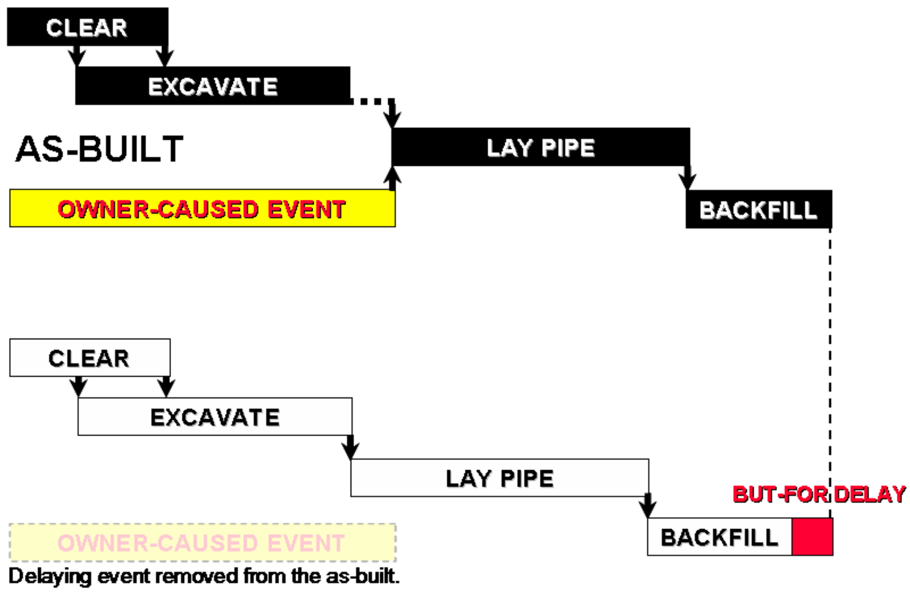


Figure 6 – Graphic Example: Modeled, Subtractive, Single Simulation

The subtractive simulation is performed on one network analysis model representing the as-built. Because it uses one network analysis model, it is technically a static logic method as opposed to a dynamic logic method. But, recall that the significance of the distinction rests in the fact that the project undergoes non-progress revisions reflecting the as-built conditions in contrast to the original baseline logic. And in view of that, a method that dynamically considers how the original logic changed is thought to be more forensically accurate than that which statically relies solely on the baseline logic. Therefore, in that context, the distinction in the case of MIP 3.8 is irrelevant since it relies on the as-built as the starting point.

MIP 3.8 is primarily used retrospectively.

B. Common Names

1. Collapsed as-built (CAB)

2. But-for analysis

3. As-built less delay

4. Modified as-built

C. Recommended Source Validation Protocols

1. Implement SVP 2.2 (as-built validation) and,

2. Implement SVP 2.4 (delay ID and quantification)

D. Enhanced Source Validation Protocols

1. Implement SVP 2.1 (baseline validation)

2. Implement SVP 2.3 (update validation)

E. Recommended Implementation Protocols

1. The as-built schedule model from which the delays are extracted is CPM logic-driven as opposed to a graphic as-built schedule. Therefore the calculated early start and early finish dates in the as-built schedule model match the actual start and actual finish dates; and, the collapsed schedule after delay extraction should also be CPM logic-driven.

2. Each change made to the as-built schedule model to create the collapsed schedule is tabulated and justified.

3. Reconcile all contract time extensions granted.

4. The as-built schedule model should contain:

    a. As-built critical path activities found in implementing Subsection 4.3 including near-critical and near-longest paths.

    b. Baseline critical path and longest path.

    c. All contractual milestones and their predecessor chains.

    d. All chains of activities alleged by the respondent to have constituted critical claimant-caused delays or concurrent delays due to specific fault of the claimant.

    e. All delays for which contract time extensions were granted.

5. The collapsing process should not involve any adjustment to logic, including lag values, or removal of constraints unless each instance of such adjustment is specifically tabulated and the basis of such adjustment explained.

6. Perform a constructability analysis of the resulting collapsed as-built schedule.

F. Enhanced Implementation Protocols

1. Reconcile the as-built and the collapsed as-built to the as-planned schedule.

2. Use all schedule activities found in the baseline schedule.

3. To account for periods during which work could not have progressed under the collapsed scenario, use a calendar simulating actual weather conditions.

G. Identification of Critical & Near-Critical Paths

Prior to the extraction of delays, pure computation of the criticality of a schedule activity under the collapsed as-built method is neither practical nor necessary. To fully verify the quantum of compensable delays and to fully account for non-compensable concurrencies, the analyst must consider and extract the delays *and then* assess the criticality of the delay. The critical path identified after the extraction process is called the analogous critical path. See Subsection 3.8.K.3

Identification of the near-controlling path at this stage is not necessary if the significant set of as-built activities were properly selected when the as-built model was prepared.

The checklist for the identification of critical and near-critical paths is as follows:

- Identify and understand all related contractual language.
- Identify the negative float theory used by the opposing party.
- If necessary, identify the as-built controlling path(s) using Subsection 4.3.C.
- After extraction of delays, identify the analogous critical path (see Subsection 3.8.K.3).

H. Identification & Quantification of Concurrent Delays & Pacing

Even in its minimum implementation, concurrency analysis is built into this method. Since the as-built, by definition, contains <u>all</u> delays that occurred on the activity paths modeled to the extent that a subset of those delays are extracted, the post-extraction schedule still contains the impact of those delays that were left in the model, thereby accounting for the concurrent impact of those delays. Because of this, often the evaluation of pacing delays is a part of the extraction process. To what extent concurrent delays are evaluated is directly related to the significant set of activities that was integrated into the as-built model.

The checklist for the identification of critical and near-critical paths is as follows:

- Determine whether compensable delay by contractor or owner is at issue.
- Identify and understand all related contractual language.
- For each delay event, distinguish the cause from the effect of delay.
- Determine whether literal or functional concurrency theory is to be used (see Subsection 4.2.).
- In a stepped extraction implementation, begin extraction with the delay event that is latest in time.
- Reconcile the total net variance between the as-built and the collapsed schedule by identifying the analogous critical path (see Subsection 3.8.K.3).
- For each suspected pacing delay event, identify the parent delay(s) and establish the order of precedence between the parent delay and the pacing delay.

- For each suspected pacing delay event, evaluate whether enough resources could have been realistically employed to perform the paced activity within its original planned duration.

I. Determination & Quantification of Excusable and Compensable Delay

Identify and understand all contractual language related to delay apportionment and determine whether the contractual language would override any determination of excusability and compensability based on findings resulting from analyses performed under this RP.

Absent such overriding language, use the following procedure.

1. Excusable & Compensable Delay (ECD)

The difference between the as-built completion date and the collapsed as-built completion date resulting from the extraction of all owner-caused delays is the total ECD. If the owner has paid the contractor specifically to accelerate, then any negative delay durations (delay mitigation) resulting from the owner-paid acceleration should be credited to the owner against the total ECD to avoid double payment to the contractor for acceleration. Where the quantification of the duration of the specific paid mitigation is not reasonably feasible, the credit adjustment may be accomplished by crediting the monetary value of the acceleration payment against the monetary value of the ECD.

2. Non-Excusable & Non-Compensable Delay (NND)

The difference between the as-built completion date and the collapsed as-built completion date resulting from the extraction of all contractor-caused delays is the total NND. If the contractor accelerated or implemented other mitigating measures and the owner did not reimburse the contractor for the cost of mitigation, the net critical mitigation duration should be subtracted from the total NND.

3. Excusable & Non-Compensable Delay (END)

Because entitlement to END does not require that concurrency periods be eliminated, this method is too rigorous for quantifying END since it automatically accounts for concurrency. However, it can be said that the difference between the as-built completion date and the collapsed as-built completion date resulting from the extraction of all owner-caused delays is *at least* the total END.

J. Identification & Quantification of Mitigation / Constructive Acceleration

The subtractive modeling methods are not the best tools for identifying and quantifying specific instances of acceleration and delay mitigation, since the methods start with the as-built schedule that already incorporates all acceleration measures to the extent that they were actually implemented. When the delays are subtracted the resulting schedule still retains all acceleration measures that were built into the as-built. Therefore, the resulting comparison is that of one accelerated schedule to another, albeit one without delays.

However, the subtractive modeling methods are one of the only tools to identify and quantify the overall extent to which the contractor's actual performance would have resulted in a project duration shorter than the baseline schedule, but for the delays. If the completion date of the collapsed schedule is earlier than that of the original baseline schedule it can be claimed by the contractor that if allowed to proceed unhindered by delays, it was possible to finish earlier than originally planned. Whether the contractor would have decided to actually incur the necessary expenses to implement the acceleratory measures absent delays must be proven independently of the schedule analysis.

K. Specific Implementation Procedures & Enhancements

1. Choice of Extraction Modes

a. Global Extraction

All the delay events and influences are extracted together and the impact is determined on the combined effect of the extracted delays.

b. Stepped Extraction

The delays are extracted individually or in groups, and the impact is determined after each iterative extraction. Stepped extraction should be in reverse chronological order of the occurrence of the delay event. This is the reverse of the order recommended for the additive MIP's 3.6 and 3.7. In the additive methods, the base schedule contains no delays, so it makes sense to start the additive process chronologically. In 3.8 the base schedule already contains all the delays. If extraction is performed chronologically, the iterative results would make no sense. For example, extracting the earliest delay first would create a schedule that still contains all the delays that occurred after the first delay.

2. Creating a Collapsible As-Built CPM Schedule

a. The first step in modeling the as-built CPM is to determine the actual duration of each schedule activity. In assigning actual durations and actual lead-lag values, use a 7-day week calendar which allows all duration units to be in calendar days rather than working days, the main reason being that often project documentation will reveal that work was performed on some days that were planned to be non-working days. The spillover advantage of using a 7-day calendar is that it significantly simplifies the reconciliation of the calculated results. This system may sometimes produce anomalous results. For example, if work started on Friday and completed on the next Monday, the duration assignment will be four days although only two were actually worked. Then in the collapse, if the same activity happens to start on the first day of a four-day holiday weekend, it will show to continue through the holiday weekend and complete on the last day of the holiday. However, the system tends to balance itself out because it is equally likely that an activity which started on a Friday and finished on the following Monday (a 2 workday activity taking 4 calendar days) would show up as occupying four workdays from a Monday through Thursday in the collapsed as-built. The counterbalancing rule is applicable to both work activities and no-work durations. Hence, the 7-day calendar is often used initially for assigning actual durations to both types of activities. Conversion to a 7-day calendar, however, may not always be appropriate. For example, when calendars include long non-work periods, such as winter breaks, it may be more appropriate to retain the original project calendars to ensure that the collapsed as-built schedule does not result in work being performed during non-work periods.

b. The as-built schedule, containing actualized data, forms the basis for creating the collapsible As-Built CPM schedule. This bar chart is modified to convert it to a CPM schedule by incorporating actual and underlying unimpacted logic relationships. The purpose of this is to allow the CPM schedule to simulate the actual activity durations and sequences solely by CPM computation using the logic ties and actual durations. The four-series diagram below illustrates this concept:

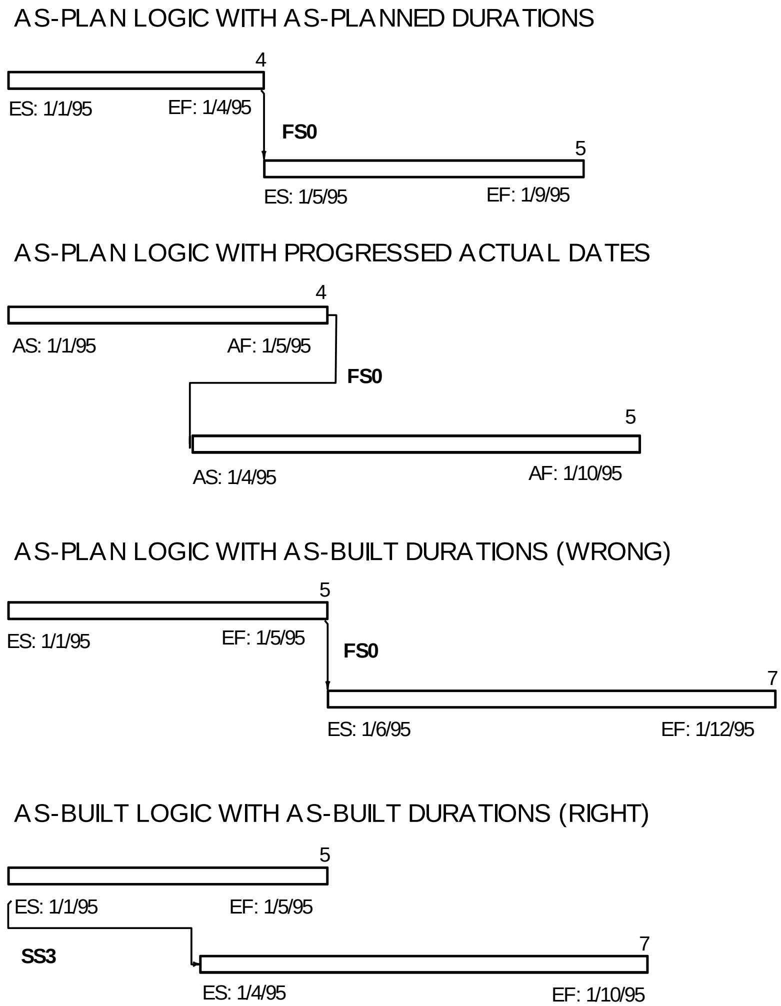


Figure 7 – Conversion of As-Planned Logic to As-Built Logic

c. Be aware that in many cases an activity should have more than one predecessor. For example, suppose that the start of wire pulling in building B was controlled by the completion of wire pulling in building A. In such a case, there would be a finish-to-start (FS) relationship with a zero lag value from “pull wire building A” to “pull wire building B”. But the installation of conduit in building B will need to be tied as a logical predecessor to wire pulling, even if that activity may not have been the controlling factor. This non-controlling relationship may become the controlling relationship if the wire pulling for building A collapses to an earlier date than conduit installation for building B.

d. Depending on the level to which the as-built logic has been developed, the activity float value in and of itself, may not be the true computed delineation of the as-built controlling path. This is illustrated in the diagrams below.

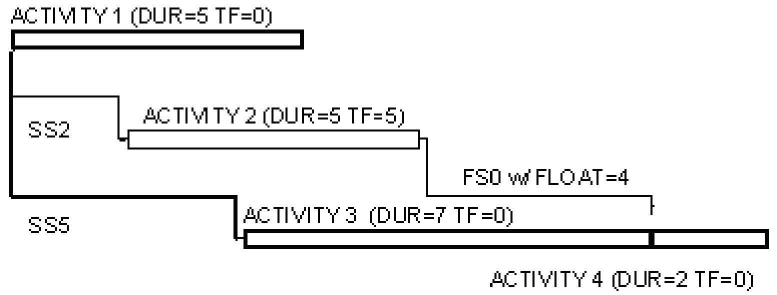


Figure 8 – As-Built Logic Showing Activity 2 Not Critical

e. The focus is on activity #2. This first model shows a FS0 logic tie from activity #2 to activity #4 allowing activity #2 to carry a float value of 5. The diagram below shows that a change to the successor logic of activity #2 to a FF5 to activity #3 will not change the dates but makes activity #2 critical.

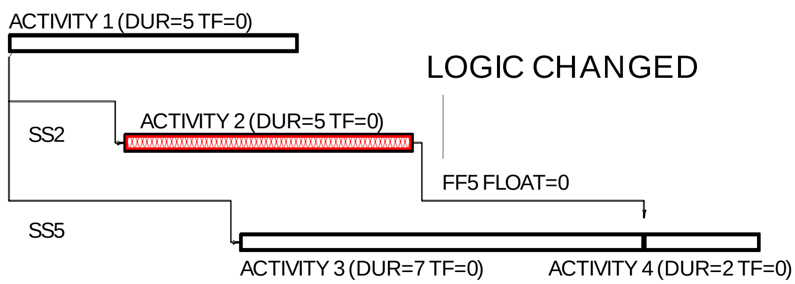


Figure 9 – Logic Change to Make Activity 2 Critical

f. Another way of looking at this FF5 logic is to model the 5 days of lag as an explicit schedule activity, and tie that to activity #4 with an FS0. While adopting a policy to replace all non-zero lag values with explicit activities and restrict all relationship ties to FS0 may simplify the logic and debugging process, it will greatly increase the number of activities to be processed.

g. If the logic change is more reflective of what actually took place, the second model is superior to the first model and is further along in the modeling process. This does not make the first model wrong because the validity of the as-built dates is intact, just the logic and the calculated float have changed. But, to rely solely on the float value of a less developed as-built model may invite error in the determination of the controlling path.

h. In most cases, simulating the actual performance of work using CPM logic requires the use of logic ties other than standard, simple, consecutive finish-to-start ties (FS0). The following is a set of guidelines to be used in assigning CPM logic ties to simulate as-built performance:

   i. Replace any FS logic with lag values 50% or longer than the duration of its predecessor or its successor, with a schedule activity.

   ii. Replace any SS Logic with lag values 50% or longer than the duration of the predecessor with a schedule activity.

iii. Replace any FF Logic with lag values 50% or longer than the duration of the successor with a schedule activity.

iv. Replace FS logic with negative lag values whose absolute value is larger than one unit of duration, with another type of logic with a zero or a positive lag that does not violate the rules stated above. Some practitioners, however, may elect to allow negative lags if the lag value is small relative to the predecessor activity duration.

v. Replace SS or FF logic with negative lag values whose absolute value is larger than one unit of duration, with another type of logic with a zero or a positive lag that does not violate the rules stated above.

vi. Where more than one type of logic tie is applicable, use the type that would use the smallest absolute lag value as the controlling logic tie.

i. This highlights the importance of this logic process, but do not expect to perfect the logic at this stage. This is due to the fact that the collapsed as-built method is most efficiently implemented as a multi-iterative process involving rapid modeling and a subsequent trial collapse which reveals faulty or incomplete as-built logic. This is repeated until the model is debugged. However, this does not excuse the analyst from using a judicious combination of expert judgment, common sense, and extensive input from project personnel with first-hand knowledge of the day-to-day events during this step of the process.

3. Identification of the Analogous Critical Path (ACP)

The analogous critical path, or ACP, is determined by transferring the calculated critical path of the collapsed as-built onto the logic path of the as-built schedule. After the delays are extracted from the as-built schedule, the remaining critical path is transferred onto the logic path of the as-built schedule. This critical path is called the analogous critical path, or ACP. The analogous critical path allows the analyst to reconcile the total delta between the collapsed state and the as-built state with the sum of those delays, whole or in part, lying on the analogous path.

Because the collapsed as-built schedule is the residual schedule after the extraction of delay activities at issue, a comparison of the critical path of the collapsed as-built with the same logic path on the as-built will yield the list of delays whose discrete durations add up to the net difference in overall duration between the two schedules.

The ACP may or may not be identical to the controlling path. The paths are identical if the sum of the delays along the controlling path is equal to the duration difference between the as-built and the collapse. A rule that can be derived from this is that the sum of delays along the ACP is equal to or less than those on the controlling path, but never more. The converse of this rule is that if a delay that does not lie on the ACP but is on the controlling path and was not extracted out of the as-built, a full collapse may not be achieved to the extent the duration of the particular delay exceeds the arithmetic difference between the sum of the delays on the ACP and the sum of all delays on the subject controlling path.

L. Summary of Considerations In Using the Minimum Protocol

- Concept is intuitively easy to understand and present

- Can isolate owner and/or contractor-caused delays if there is sufficient detail in the as-built schedule.
- Relies upon history of actual events.
- Can be implemented without any baseline schedule or contemporaneous schedule updates.
- Relatively few practitioners with significant, hands-on experience in properly performing this method.

M. Caveats In Using the Minimum Protocol / Conditions Requiring Enhanced Protocols

- Perceived to be purely an after-the-fact reconstruction of events that does not refer to schedule updates used during the project
- Summarized as-built variation of the minimum protocol creates the potential for missing scope of work or the skewing of results of the analysis.
- Reconstructing the as-built schedule is very fact and labor intensive.
- Assignment of logic to mimic as-built conditions requires subjective decisions that sometimes do not match the contemporaneously planned logic relationships between activities.
- Indicated as-built critical path throughout project does not necessarily reflect changes in the prospective critical path indicated in contemporaneous schedule updates.
- Susceptible to unintended or intended manipulation during as-built logic assignments.
- May ignore prospective critical paths projected in the contemporaneous schedule updates along with the project management decisions that were based upon those critical paths
- Not suited for identification or quantification of acceleration because the source as-built schedule already incorporates acceleration.

3.9. Modeled / Subtractive / Multiple Base (MIP 3.9)

A. Description

Like MIP 3.8,MIP 3.9 is a modeled technique relying on a simulation of a CPM model scenario. The simulation consists of the extraction of entire activities or a portion of the as-built durations representing delays or changes from a network analysis model representing the as-built condition of the schedule to determine the impact of those extracted activities to each network model. Hence, MIP 3.9 is also a subtractive model.

MIP 3.9 is a multiple base method, distinguished from MIP 3.8 which is a single base method. The subtractive simulation is performed on multiple network analysis models representing the as-built schedule, typically updated schedules, which may include contemporaneous, modified contemporaneous, or recreated schedules. As the project undergoes non-progress revisions in reaction to the as-built conditions, in contrast to the original baseline logic, MIP 3.9 considers those logic changes and, therefore, is thought to be more attuned to the perceived critical path, in addition to the actual critical path that existed during the project than methods which rely solely on the initial baseline or the final as-built. Because the updates typically include non-progress revisions, MIP 3.9 is a dynamic logic method as opposed to a static logic method.

The subtractive simulation is performed on periodic network analysis models representing intervals of the as-built schedule. Each model creates a time period of analysis that confines the quantification of delay impact. Forecasted delays beyond an analysis period, however, may also need to be extracted at the time that the forecasted delays are introduced into the schedule. For example, a schedule update may include a change order impact inserted into the update to forecast delay events which is expected to occur several months after the schedule update period. This may distort the delay calculations when compared with the previous schedule being used as the baseline for the analysis. Thus, these forecasted impacts may need to be removed from the analysis period under consideration in order to properly quantify current impacts.

MIP 3.9 shares an important technical consideration with MIP 3.5 (Observational / Dynamic / Modified or Recreated), namely the choice in using hindsight or blindsight in recreating, and in the case of MIP 3.9, modeling activities that were partially complete on a given data date.

MIP 3.9 is primarily used retrospectively.

B. Common Names

1. Collapsed As-Built (CAB)

2. Windows Collapsed As-Built

3. But-For Analysis

4. Windows As-Built But-For

5. As-Built Less Delay

6. Modified As-Built

7. Look-Back Window

C. Recommended Source Validation Protocols

1. Implement SVP 2.2 (as-built validation),

2. Implement SVP 2.3 (update validation) and,

3. Implement SVP 2.4 (delay ID and quantification)

D. Enhanced Source Validation Protocols

1. Implement SVP 2.1 (baseline validation)

E. Recommended Implementation Protocols

1. The as-built schedule models from which the delays are extracted are CPM logic-driven as opposed to graphic as-built schedules. Therefore the calculated early start and early finish dates in the as-built schedule models match the actual start and actual finish dates and the collapsed schedules after delay extraction should also be CPM logic-driven.

2. Each change made to the as-built portion of the schedule for each time period to create the collapsed schedule is tabulated and justified.

3. There should be **at least** two base models, consisting of one based on a partially progressed schedule update and a second one based on a fully progressed schedule update or an as-built schedule.

4. The as-built schedule models should contain:

    a. As-built critical path activities found in implementing Subsection 4.3 including near-critical and near-longest paths.

    b. Baseline critical path and longest path.

    c. All contractual milestones and their predecessor chains.

    d. All chains of activities alleged by the respondent to have constituted critical claimant-caused delays or concurrent delays due to specific fault of the claimant.

    e. All delays for which contract time extensions were granted.

5. The collapsing process should not involve any adjustment to logic, including lag values, or removal of constraints unless each instance of such adjustment is specifically tabulated and the basis of such adjustment explained.

6. Perform a constructability analysis of the resulting collapsed as-built schedules.

7. Reconcile all contract time extensions granted.

F. Enhanced Implementation Protocols

1. Reconcile the as-built and the collapsed as-built to the as-planned schedule.

2. Use all schedule activities found in the baseline schedule.

3. To account for periods during which work could not have progressed under the collapsed scenario, use a calendar simulating actual weather conditions.

4. Perform the analysis by modeling <u>all</u> schedule updates.

5. For each time period, create two models, one using hindsight progress rules, and the other using blindsight progress rules in modeling activities that were partially complete on the data date.

G. Identification of Critical & Near-Critical Paths for Each Periodic Update

Prior to the extraction of delays, pure computation of the criticality of a schedule activity under the collapsed as-built method is neither practical nor necessary. To fully verify the quantum of compensable delays, and to fully account for non-compensable concurrencies, the analyst must consider and extract the delays *and then* assess the criticality of the delay. This analogous critical path is used to identify the controlling activities of the collapsed as-built. See Subsection 3.9.K.5

Identification of the near-controlling path at this stage is not necessary if the significant set of as-built activities were properly selected when the as-built model was prepared.

The checklist for the identification of critical and near-critical paths is as follows:

- Identify and understand all related contractual language.

- Identify the negative float theory used by the opposing party.

- Identify and understand the implications of the choice of method, hindsight or blindsight, when modeling remaining durations of partially complete activities. (See Subsection 4.2.D.6)

- If necessary, identify the as-built controlling path(s) using Subsection 4.3.C.

- After extraction of delays, identify the analogous critical path (ACP). (See Subsection 3.9.K.5)

H. Identification & Quantification of Concurrent Delays & Pacing

As with MIP 3.8, even in its minimum implementation, concurrency analysis is built into MIP 3.9. Since the as-built, by definition, contains <u>all</u> delays that occurred on the activity paths modeled, to the extent that a subset of those delays are extracted, the post-extraction schedule still contains the impact of those delays that were left in the model, thereby accounting for the concurrent impact of those delays. Because of this, often the evaluation of pacing delays is a part of the extraction process. To what extent concurrent delays are evaluated is directly related to the significant set of activities that were integrated into the as-built model. However, the analyst must be aware that unlike MIP 3.8, this method contains a retrospective and a prospective portion within the logic-driven portion of each model. (See Figure 10).

The checklist for the identification of critical and near-critical paths is as follows:

- Determine whether compensable delay by contractor or owner is at issue.

- Identify and understand all related contractual language.

- For each delay event, distinguish the cause from the effect of delay.

- Determine whether literal or functional concurrency theory is to be used (see Subsection 4.2.).

- In a stepped extraction implementation, begin extraction with the delay event that is latest in time in the period being analyzed.

- Reconcile the total net variance between the as-built and the collapsed schedule by identifying the analogous critical path. (See Subsection 3.9.K.5)

- For each suspected pacing delay event, identify the parent delay(s) and establish the order of precedence between the parent delay and the pacing delay.

- For each suspected pacing delay event, evaluate whether enough resources could have been realistically employed to perform the paced activity within its original planned duration.

I. Determination & Quantification of Excusable and Compensable Delay

Identify and understand all contractual language related to delay apportionment and determine whether the contractual language would override any determination of excusability and compensability based on findings resulting from analyses performed under this RP.

Absent such overriding language, use the following procedure.

1. Excusable & Compensable Delay (ECD)

The difference between the as-built completion date and the collapsed as-built completion date resulting from the extraction of all owner-caused delays is the total ECD for each modeled time period. If the owner has paid the contractor specifically to accelerate, then any negative delay durations (delay mitigation) resulting from the owner-paid acceleration should be credited to the owner against the total ECD to avoid double payment to the contractor for acceleration. Where the quantification of the duration of the specific paid mitigation is not reasonably feasible, the credit adjustment may be accomplished by crediting the monetary value of the acceleration payment against the monetary value of the ECD.

2. Non-Excusable & Non-Compensable Delay (NND)

The difference between the as-built completion date and the collapsed as-built completion date resulting from the extraction of all contractor-caused delays is the total NND for each modeled time period. If the contractor accelerated or implemented other mitigating measures and the owner did not reimburse the contractor for the cost of mitigation, the net critical mitigation duration should be subtracted from the total NND.

3. Excusable & Non-Compensable Delay (END)

Because entitlement to END does not require that concurrency periods be eliminated, this method is too rigorous for quantifying END since it automatically accounts for concurrency. However, it can be said that the difference between the as-built completion date and the collapsed as-built completion date resulting from the extraction of all owner-caused delays is _at least_ the total END for each modeled time period.

.

J. Identification & Quantification of Mitigation / Constructive Acceleration

The subtractive modeling methods are not the best tools for identifying and quantifying specific instances of acceleration and delay mitigation, since the methods start with the as-built schedule that already incorporates all acceleration measures to the extent that they were actually implemented. When the delays are subtracted, the resulting schedule still retains all acceleration measures that were built into the as-built. Therefore, the resulting comparison is that of one accelerated schedule to another, albeit one without delays.

However, the subtractive modeling methods are one of the only tools to identify and quantify the overall extent to which the contractor's actual performance would have resulted in a project duration shorter than the baseline schedule, but for the delays. If the completion date of the collapsed update is earlier than that of the schedule update of the previous period, it can be claimed by the contractor that if allowed to proceed unhindered by delays, it was possible to finish earlier than originally planned. Whether the contractor would have decided to actually incur the necessary expenses to implement the acceleration measures absent delays must be proven independently of the schedule analysis.

K. Specific Implementation Procedures & Enhancements

1. Choice of Analysis Periods

a. Fixed Periods

The analysis periods are of virtually identical duration and may coincide with regular schedule update periods.

b. Variable Periods

The analysis periods are of varying duration and are characterized by their different natures such as the type of work being performed, the types of delaying influences, or the operative contractual schedule under which the work was being performed.

c. Fixed-Periods vs. Variable-Periods

Similar to the comparison between the all-periods implementation and the grouped-periods implementation for MIP 3.3, 3.4, and 3.5, a frequent-fixed-periods implementation yields more information than the infrequent-variable-periods implementation and is considered more precise.

2. Choice of Modeling Increments

a. Periodic Modeling

In periodic modeling, the logic-driven as-built schedule occupies the period starting with the day after the data date of the previous update and ending with the data date of the current update from which the as-built model is generated. The data date of the previous update remains the data date for the model. This data date will be referred to as the hard data date of the model in order to distinguish it from the soft data date which is the data date of the current update from which the model was generated. The soft data date is so named because the calculation discontinuity of the data date of the source update is blurred or softened in the continuous CPM logic spanning the source update data date, as shown in the diagram below.

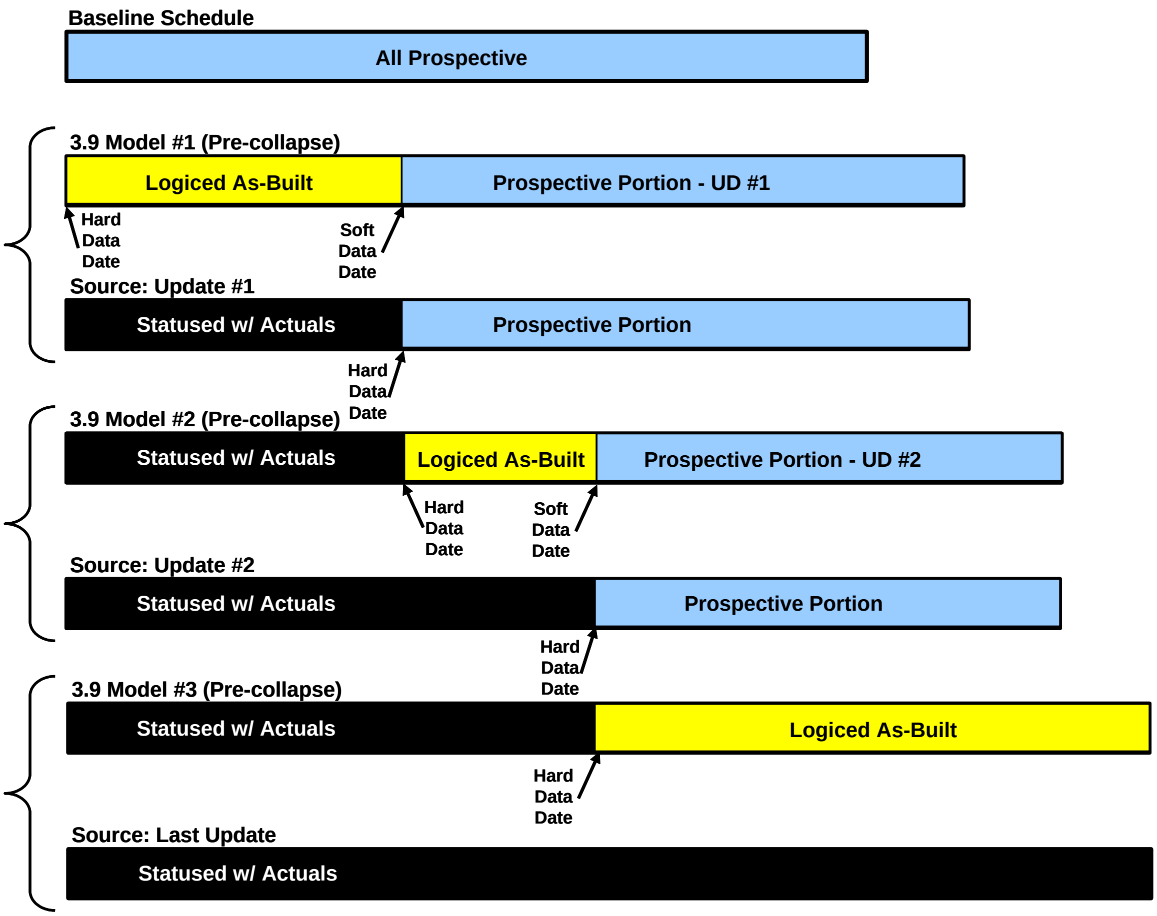


Figure 10 – Graphic Example: MIP 3.9 With Periodic Modeling

Hindsight progress rules are used to model the as-built at the hard data date of the model, since this point in time is already fully progressed in the source update. The analyst has a choice of rules, hindsight or blindsight, in modeling the as-built at the soft data date since on one hand, this point in time is the hard data date of the source update, but on the other hand, if the analysis is being performed after project completion, full as-built information is available. The difference in progress rules used for modeling may make a difference in the calculation of the critical path(s), near-critical paths, longest path(s), and the near-longest paths.

b. Cumulative Modeling

In a cumulatively modeled set of MIP 3.9 as-builts, the hard data date is set for the first model, and all subsequent models use the same hard data date. In many cases the initial hard data date is the same as that of the baseline schedule. The soft data date of the models moves with the data date of the source updates. If the final source update is a fully progressed update, the final as-built model will be identical to a MIP 3.8 model based on a fully progressed update, as shown in the diagram below.

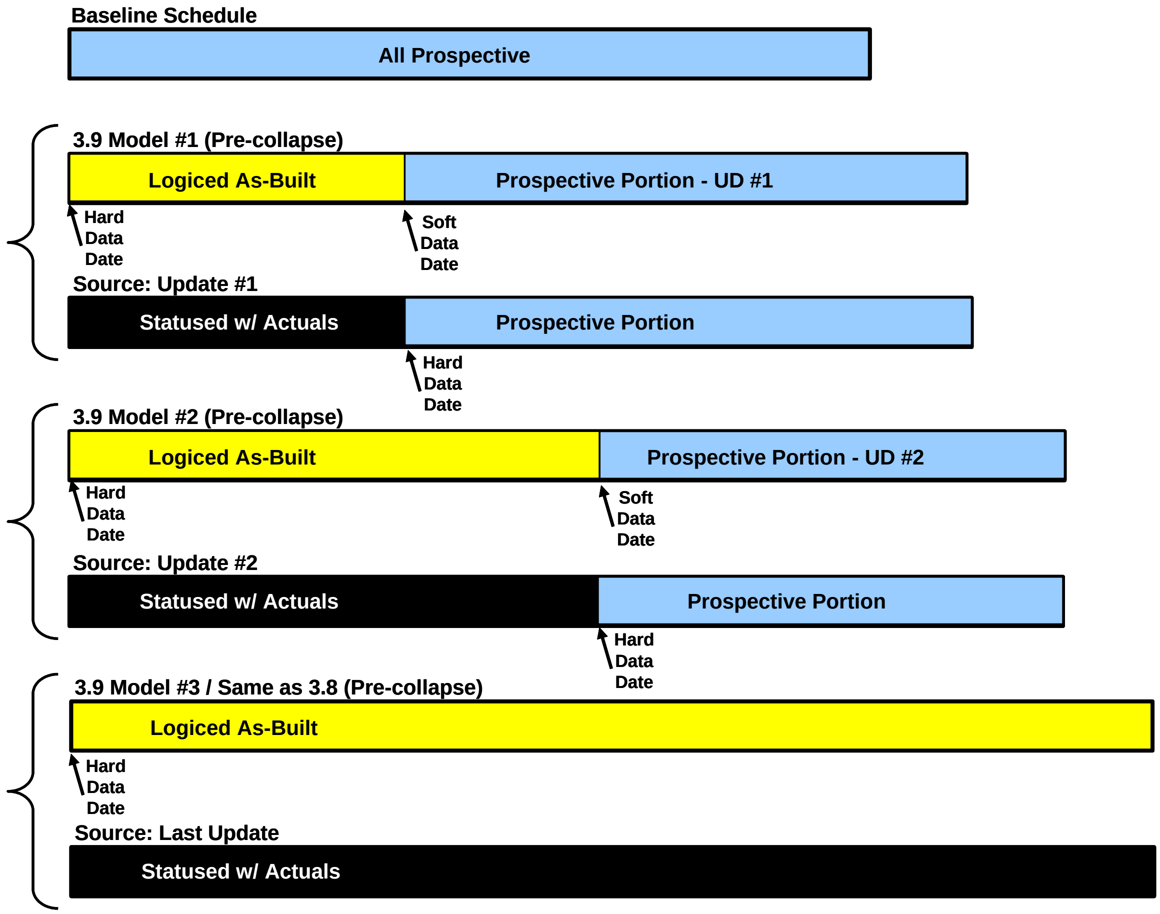


Figure 11 – Graphic Example: MIP 3.9 With Cumulative Modeling

As with the periodic modeled set of as-builts, the analyst has a choice of rules, hindsight or blindsight, in modeling the as-built at the soft data date since on one hand, this point in time is the hard data date of the source update, but on the other hand, if the analysis is being performed after project completion, full as-built information is available. The difference in progress rules used for modeling may make a difference in the calculation of the critical path(s), near-critical paths, longest path(s), and the near-longest paths.

3. Choice of Extraction Modes

a. Global Extraction

All the delay events and influences in each model are extracted together and the impact is determined on the combined effect of the extracted delays.

b. Stepped Extraction

The delays are extracted individually or in groups, and the impact is determined after each iterative extraction. Stepped extraction should be in reverse chronological order of the occurrence of the delay event. This is the opposite of the order recommended for the additive MIP's, 3.6 and 3.7. In the additive methods, the base schedule contains no delays, so it makes sense to start the additive process chronologically. In MIP 3.9, the base schedules already contain all the delays. If extraction is performed chronologically, the iterative results would make no sense. For example, extracting the earliest delay first would create a schedule that still contains all the delays that occurred after the first delay.

4. Creating a Collapsible As-Built CPM Schedule

The procedure for creating a collapsible as-built schedule for each period analysis is the same as presented in Subsection 3.8.K.2, except that the process must be repeated for the relevant analysis period for each as-built schedule update.

5. Identification of the Analogous Critical Path (ACP)

The procedure for identifying the Analogous Critical Path for each period analysis is the same as presented in Subsection 3.8.K.3, except that the process must be repeated for the relevant analysis period for each as-built schedule update.

L. Summary of Considerations In Using the Minimum Protocol

- Accounts for changes in the prospective critical path for each schedule update utilized
- Concept is intuitively easy to understand and present
- Can isolate owner and/or contractor-caused delays if there is sufficient detail in the as-built schedule.
- Relies upon history of actual events.
- This method requires a baseline schedule and subsequent schedule updates in addition to the as-built schedule.
- Relatively few practitioners with significant, hands-on experience in properly performing this method.

M. Caveats In Using the Minimum Protocol / Conditions Requiring Enhanced Protocols

- Summarized as-built variation of the minimum protocol creates the potential for missing scope of work or the skewing of results of the analysis.
- Reconstructing the as-built schedule is very fact and labor intensive.
- Assignment of logic to mimic as-built conditions requires subjective decisions that sometimes do not match the contemporaneously planned logic relationships between activities.
- Susceptible to unintended or intended manipulation during as-built logic assignments.

- Not suited for identification or quantification of acceleration because the source as-built schedule already incorporates acceleration.

- More time-consuming and hence more expensive to implement than other MIP's.

4. ANALYSIS EVALUATION

4.1 Excusability and Compensability of Delay
4.2 Identification and Quantification of Concurrency of Delay
4.3 Critical Path and Float
4.4 Delay Mitigation & Constructive Acceleration

The ultimate conclusion sought in forensic schedule analysis involving delay disputes is the determination and quantification of excusable delays along with the compensability of such delays. The analysis methods outlined in Section 3 are the tools used in reaching this ultimate conclusion[7]. This section describes the procedures for interpreting the results obtained from the use of the methods described in Section 3.

The process of segregating non-excusable, excusable, and compensable delays is referred to herein as *apportionment* of the responsibility for delay. Many jurisdictions in the United States and other countries prefer the use of critical path method (CPM) techniques for the purpose of apportionment of delay. This is in distinction to the use of other techniques such as bar-charts without network logic or by gross allocation of fault by percentage, often called the pie-chart method.

Subsection 4.1 was placed first so that the reader can gain an overview before delving into the underlying technical concepts. The analyst must be familiar with the concepts of concurrency of delay (Subsection 4.2), and criticality and float (Subsection 4.3) in order to fully understand the concepts in the first Subsection, 4.1. Therefore, for issues involving delay, the actual order of performance of the analysis interpretation protocol would be Subsection 4.3 first, then 4.2 followed by 4.1.

Constructive acceleration, along with recovery schedules, disruption, and delay mitigation are addressed in Subsection 4.4. Even if the project did not result in actual slippage of the completion date, these issues still generate disputes. Because the issues are intertwined with excusability of delay, they are discussed here in Section 4.

Be advised that differences in analysis methods combined with differences in concurrency and float theories may result in conflicting ultimate conclusions. The primary purpose of this section is to describe and explain the different theories in order to aid in the reconciliation of the conflicting conclusions.

4.1. Excusability and Compensability of Delay

A. General Rules[8]

Excusability exists where there is contractual or equitable justification in a claimant's request for a contract time extension for relief from potential claims for liquidated/stipulated or actual delay damages. The showing of excusability does not necessarily mean that the claimant is also entitled to compensation for the delay. Conversely, delay is *non-excusable* when such justification does not exist.

[7] As a practical matter, delay analysis is just an intermediate step towards the ultimate question of financial liability. Thus, if agreement can be reached directly on the question of the specific amount of financial liability, the forensic schedule analysis leading to an apportionment of delay liability is moot.
[8] The contracting parties are free to depart from the general rule by mutual agreement as long as such agreement does not violate public policy.

Compensability or *compensable delay* exists where the claimant is entitled to recover not only a time extension but compensation for expenses associated with the extension of completion date or the prolongation of the duration of work. Excusability is a prerequisite to compensability. Therefore, where compensability can be established, excusability is assumed.

B. Accounting for Concurrent Delay

In the absence of any contractual language or other agreements, the conventional rule governing compensability is that the claimant must first account for concurrent delays (see Subsection 4.2) in quantifying the delay duration to which compensation applies. That is, the contractor is barred from recovering delay damages to the extent that concurrent contractor-caused delays offset owner-caused delays, and the owner is barred from recovery liquidated/stipulated or actual delay damages to the extent that concurrent owner-caused delays offset contractor-caused delays.

The evaluation proceeds in two distinct steps. First, the liability for each delay event is individually analyzed[9]. The classification is made primarily according to the responsibility for the cause of the delay but may also consider the contractual risk allocation of the delay event regardless of the party who caused such delay. The second step consists of evaluating whether each delay event is concurrent with other types of delays to arrive at the final conclusion of excusability, compensability, or non-excusability.

As evident from the list of existing definitions, the current, common usage of the terms compensable, excusable, and non-excusable is confusing because analysts often use those terms to characterize the assignment of liability performed in the first step. For the purpose of this RP, the delays identified in the first step will be classified as: contractor delay, owner delay, or *force majeure* delay.

A *contractor delay* is any delay event caused by the contractor or the risk of which has been assigned solely to the contractor[10]. If the contractor delay is on the critical path, in the absence of other types of concurrent delays, the contractor is granted neither an extension of contract time nor additional compensation for delay related damages. Such a delay may expose the contractor to a claim for damages from the owner.

An *owner delay* is any delay event caused by the owner, or the risk of which has been assigned solely to the owner[11]. If the owner delay is on the critical path, in the absence of other types of concurrent delays, the contractor is granted both an extension of contract time and additional compensation for delay related damages.

A *force majeure delay* is any delay event caused by something or someone other than the owner (including its agents), or the contractor (or its agents), or the risk of which has not been assigned solely to the owner or the contractor. If the *force majeure* delay is on the critical path, the contractor is granted an extension of contract time but does not receive additional compensation for delay related damages even if there is a concurrent delay.

[9] Note that the forensic scheduling analyst may not possess the skill, knowledge, or experience to independently determine the legal liability for an event. In such a case, the first step consists of making a reasoned assumption of liability subject to verification by those with the requisite expertise.
[10] The SCL Delay & Disruption Protocol calls this a contractor risk event which is defined as an event or cause of delay which under the contract is at the risk and responsibility of the contractor. SCL also calls it a non-compensable event.[1]
[11] The SCL Delay & Disruption Protocol calls this an employer risk event which is defined as an event or cause of delay which under the contract is at the risk and responsibility of the employer (owner). SCL also calls it a compensable event.[1]

After liability is determined in the first step, the second step requires a determination of concurrency in accordance with Subsection 4.2. The various permutations of concurrency scenarios are summarized below in Figure 12 – *Net Effect Matrix*.

Delay Event	Concurrent with	Net Effect
Owner Delay	Another Owner Delay or Nothing	Compensable to Contractor, Non-Excusable to Owner
Owner Delay	Contractor Delay	Excusable but Not Compensable to both Parties
Owner Delay	Force Majeure Delay	Excusable but Not Compensable to both Parties
Contractor Delay	Another Contractor Delay or Nothing	Non-Excusable to Contractor, Compensable to Owner
Contractor Delay	Force Majeure Delay	Excusable but Not Compensable to both Parties
Force Majeure Delay	Another Force Majeure Delay or Nothing	Excusable but Not Compensable to Contractor

Figure 12 – Net Affect Matrix – Concurrent Delay

There are two alternatives if there are more than two parties among which the delay must be apportioned depending on whether the additional parties are distinct signatories to the subject contract or whether the parties are agents and therefore subsumed under the two primary parties.

Under the first alternative there would be another factor added to the above matrix. But, the principle used to derive the net effect would be the same. Namely, in order to be entitled to compensation the party must not have caused or otherwise be held accountable for any concurrent delay and concurrent *force majeure* delays.

Under the second alternative involving agents to the two primary parties such as subcontractors, suppliers, architects, and construction management firms, the net effect equation should be solved first between the two primary parties. This is followed by a subsidiary analysis apportioning the quantified delay allocation established by the first analysis.

C. Equitable Symmetry of the Concept

Note that the terms compensable, excusable, and non-excusable in current industry usage are from the viewpoint of the contractor. That is, a delay that is deemed compensable is compensable to the contractor but non-excusable to the owner. Conversely, a non-excusable delay is a compensable delay to the owner since it results in the collection of liquidated/stipulated damages.

A neutral perspective on the usage of the terms often aids understanding of the parity and symmetry of the concepts[12]. Thus entitlement to compensability, whether it applies to the contractor or the owner, requires that the party seeking compensation shows a lack of concurrency. But for entitlement to excusability without compensation, whether it applies to

[12] Especially in the absence of contractual provisions to the contrary. For example, depending on the contract language and applicable law, the applicable tests for the recovery of actual delay damages may be different from that applicable to the owner's right to liquidated/stipulated damages.

the contractor or the owner, it only requires that the party seeking excusability show that a delay by the other party impacted the critical path.

Based on this symmetry, contractor entitlement to a time extension does not automatically entitle the contractor to delay compensation. In addition to showing that an owner delay impacted the critical path, the contractor would have to show the absence of concurrent delays caused by a contractor delay or a *force majeure* delay in order to be entitled to compensation.

A contractor delay concurrent with many owner delays would negate the contractor's entitlement to delay compensation. Similarly, one owner delay concurrent with many contractor delays would negate the owner's entitlement to delay compensation, including liquidated/stipulated damages. While in such extreme cases the rule seems draconian, it is a symmetrical rule that applies to both the owner and the contractor and hence ultimately equitable.

4.2. Identification and Quantification of Concurrent Delay

A. Relevance and Application

Projects are frequently delayed by multiple impacts and by multiple parties. The concept of concurrent delay is based upon the premise that when multiple parties independently contribute to an impact to the critical path, the party or parties causing the event should be responsible for their share of that project critical path impact. There can be concurrent delays between separate delay events both caused by the same party. However, in such case there is effectively no need for a concurrency analysis. Throughout this Recommended Practice, it has been assumed that concurrency exists only when it is caused by at least two separate parties or between at least one party and a force majeure event. While the allocation and distribution of concurrent delay impacts should always be based upon the terms and conditions of the contract, most contracts are silent on the subject of concurrent delay. This section is intended to identify and facilitate the calculation and apportionment of concurrent delay impacts.

Typically, Owners assess liquidated/stipulated damages for non-excusable delay and Contractors claim entitlement to extended overhead reimbursement for compensable delay. In each case, the damages are typically calculated on the basis of a daily unit rate. Under most concurrent delay applications however, the Owner and Contractor time-related damages are not offset against each other when concurrent delay can be demonstrated. Typically, when both Contractor and Owner are concurrently responsible for an extended period of performance, the Contractor is granted an extension of contract without compensation and the Owner forgoes the collection of liquidated/stipulated damages. No time-related compensation flows from either party to the other. Generally, therefore, substantial incentive exists for:

1. The Contractor to demonstrate concurrent excusable delay during a period likely to be considered non-excusable delay; and
2. The Owner to demonstrate concurrent non-excusable delay during a period likely to be considered excusable delay.

Accordingly, both Owners and Contractors frequently contend that concurrent delays offset each other as a defense to excuse their potential liability to compensate the other party for time related costs.

The identification and quantification of concurrent delay is arguably the most contentious technical subject in forensic schedule analysis. Accordingly, it is important that all sides, if possible, agree on either the Literal or Functional theory (See Subsection 4.2.D.1.) employed in the identification and quantification of concurrent delay. Failing that, the analyst should be aware of the theory adopted by the opposing party.

B. Various Definitions of Concurrency

AACE RP10S-90 "Cost Engineering Terminology," lists five different but similar definitions for concurrent delay.[4] As discussed more fully in the sections that follow, the five definitions reflect some of the differing opinions and applications associated with concurrent delay. The apparent contradictions underscore why this has become one of the most contentious areas of forensic schedule delay analysis.

(1) Two or more delays that take place or overlap during the same period, either of which occurring alone would have affected the ultimate completion date. In practice, it can be difficult to apportion damages when the concurrent delays are due to the owner and contractor respectively.

(2) Concurrent delays occur when there are two or more independent causes of delay during the same time period. The "same" time period from which concurrency is measured, however, is not always literally within the exact period of time. For delays to be considered concurrent, most courts do not require that the period of concurrent delay precisely match. The period of "concurrency" of the delays can be related by circumstances, even though the circumstances may not have occurred during exactly the same time period.

(3) True concurrent delay is the occurrence of two or more delay events at the same time, one an employer risk event, the other a contractor risk event and the effects of which are felt at the same time. The term 'concurrent delay' is often used to describe the situation where two or more delay events arise at different times, but the effects of them are felt (in whole or in part) at the same time. To avoid confusion, this is more correctly termed the 'concurrent effect' of sequential delay events.

(4) Concurrent delay occurs when both the owner and contractor delay the project or when either party delays the project during an excusable but non-compensable delay (e.g., abnormal weather). The delays need not occur simultaneously but can be on two parallel critical path chains.

(5) The condition where another delay-activity independent of the subject delay is affecting the ultimate completion of the chain of activities.

The existence of a contractual definition is a major factor on the determination of concurrency. As stated in the previous subsections, contracting parties are free to mutually agree on any method or procedure as long as those agreements are legally enforceable. Therefore, the general rules, exceptions, and considerations in this RP are applicable to the extent that they do not directly contradict contractual definitions and specifications.

C. Pre-requisite Findings Concerning the Delays Being Evaluated for Concurrency

Before evaluation of concurrency, there must be:

- Two or more delays that are unrelated, independent, and would have delayed the project even if the other delay did not exist;
- Two or more delays that are the contractual responsibility of different parties, but one may be a force majeure event.;
- The delay must be involuntary;

- The delayed work must be substantial and not easily curable.

1. **Two or more delays that are unrelated and independent**.

Concurrent delays occur when two or more unrelated and independent events delay the project. When two or more parties contribute to a single delay to the project and the causation is linked or related, the event is not considered to have two concurrent causes. The distinction between concurrent delay and mutually-caused delay is a subtle, yet a vitally important distinction that each analyst must observe and reconcile.

There must be at least two independent delay events. The first event, for example, could be the Owner's failure to timely approve the purchase of a piece of Owner-furnished equipment. The second and potentially concurrent event could be the Contractor's failure to advance steel erection sufficiently to support the installation of that equipment. These two independent events are often separate, co-critical network paths, but they need not be in order to be candidates for a concurrent delay. The delay events could affect the same activity, but must be independent.

Care must be taken to ensure the events are truly independent. In the example above, the facts *might* show that the steel was not erected timely because the Contractor knew the equipment was going to be late. In such a case, the "two" delay events are actually one – they are both caused by the Owner's failure to timely approve the purchase of a piece of equipment.

2. **Two or more delays that are the contractual responsibility of different parties.**

The application of concurrent delay theory is only relevant when the delays are the responsibility of different parties or one of the delays is a force majeure event. Since the concept of concurrency has both a legal and a technical component, the concurrent events must contractually be the responsibility of separate parties. The parties are typically the Owner and the Contractor. Some contracts contain language assigning responsibility or contractual risk for certain types of events such as differing site conditions and force majeure events. Such risk assignment may impact the liability of events causing concurrent delay.

If one of the delay events is contractually assigned to neither or both parties, such as a force majeure event, the effective result is the same as concurrency; it is excusable and non-compensable to either party. Generally, whenever a force majeure event occurs, it *trumps* any other concurrent delay that might have occurred. This serves two purposes: first, it can eliminate or reduce significant proof problems that might arise in establishing responsibility, and second, it promotes equity, since one of the delays is beyond the control and responsibility of the either of the parties.

3. **The delay must be involuntary**

A delay that otherwise meets the requirements of concurrency, but is performed voluntarily is generally considered pacing. If the delay could have been easily cured, but was not, the delay would be considered voluntary. See Subsections 4.2 E and F below.

4. **The delay must be substantial and not easily curable.**

This requirement comports with common sense. If one of the delays is associated with a minor element of work that could easily be performed, that work should not create a concurrent delay. This element is closely allied with the involuntary nature of truly concurrent delays cited above.

D. Functional Requirements Establishing Concurrency and the Factors that Influencing Findings

Having satisfied the four requirements on the nature of the subject delay events being evaluated for concurrency, there are two major functional requirements relating to the relationship of the delays.

- The delays must occur during or impact the same time analysis period.
- The delays, each of which, absent the other, must independently delay the critical path.

The first functional requirement that the delays must occur during or impact the same analysis time period is intuitively obvious, but difficult to absolutely satisfy. This is due to the fact that absolute, literal concurrency is an unachievable goal since time is infinitely divisible. It is more a function of the planning unit used by the schedule or the verification unit used in the review of the as-built data. For example, upon further examination, a pair of events that were determined to have occurred concurrently on a given day may not be literally concurrent because one occurred in the morning and the other in the afternoon. This condition seldom occurs since most construction schedules use the day as the smallest measurement of time.

The second functional requirement is that each concurrent delay event must, absent the other, delay the timely completion of a completion milestone. Such independent events must also be on the critical path or near critical path, depending on the time analysis period and the concurrency theory being used. For example, assume that a forensic analysis confirms that the late installation of drywall caused a critical path delay to the completion of the project. This work was critical to the commencement of final painting and interior trim work. Further assume that the delay in the drywall was the result of two factors: first, the general contractor failed to procure its drywall subcontractor in a timely manner and second, there was a severe shortage of drywall to the region. These events are unrelated, but either one of them would have delayed the overall completion of the drywall. This test is sometimes called the “but-for” test. But-For the failure to procure the drywall subcontractor, the work would still have been late because of the shortage of materials.

Findings of concurrency analysis to determine compliance with these functional requirements are highly dependent on several factors, all of which are dictated by discretionary choices made by the analyst in the course of analysis – these choices should be well documented as part of the analysis. There are at least six factors, each discussed in detail below, that influence the determination of these two conditions:

- Whether concurrency is determined literally or functionally
- Whether criticality is determined on least-value float or less-than-one float value
- Whether concurrency is determined on the cause or the effect of delay
- The frequency, duration and placement of the analysis interval
- The order of delay insertion or extraction in a stepped implementation
- Whether the analysis is done using full hindsight or blindsight (knowledge-at-the-time).

There is no consensus on the many factors that affect the identification and quantification of concurrency. The one thing that seems to be universally accepted is that reliable identification and quantification of concurrency must be based on CPM concepts, particularly distinguishing critical from non-critical delays. Gross concurrency, or the method of counting concurrent delay events based purely on contemporaneous occurrence without regard to CPM principles, is typically not a sufficient basis for concluding that a delay is not compensable.

1. Literal Concurrency vs. Functional Concurrency

There are two different theories regarding the exact timing of the two or more delays that are candidates for concurrency. Under the Literal Theory, the delays have to be literally concurrent in time, as in "happening at the same time." In contrast, under the Functional Theory, the delays need to be occurring within the same analysis period.

Of the two, the functional theory is more liberal in identifying and quantifying concurrency since the delays need only occur within the same measurement period, while in the literal theory, only delays require same-time occurrence. The assumption made by the functional theory practitioner is that most delays have the potential of becoming critical, once float on the path on which they resides has been consumed.

An advocate of *functional concurrency* believes that if the two delays occur within the same measurement period [usually a month] they can be concurrent. For example, analyses that are based upon monthly update submissions will manifest delay only at the end of the month. It is quite possible therefore, that an Owner-caused delay occurring in the first week of the update period may appear concurrent with a Contractor-caused delay occurring in the last week of the update period. These delay events could nonetheless be concurrent so long as the other tests are met. Accordingly, the functional application of concurrent delay theory does not necessarily require the delay events to occur on the same days.

This type of functional concurrency is closely attuned to delay methodologies that use modeled CPM schedules as their basis and utilize some form of time period analysis. Since these analyses measure delay at the end of time periods [typically the status updates] it makes sense to measure concurrency under this methodology at the same points, rather than trying to develop a separate concurrency analysis. Accordingly, the functional application of concurrent delay theory does not necessarily require the delay events to occur at the same time. In addition, the functional theory allows that CPM schedules, even if properly maintained, are not perfect, and near critical delays may in fact be concurrent.

The literal theory will result in the identification of fewer concurrent delays, since delays are dropped from the list of suspects if they do not share real-time concurrency. Since the literal theory is based on the general notion that concurrent delays must be on the critical path and occur at the same time (usually measured at a day interval), findings of concurrency are exceedingly rare.

An advocate of *literal concurrency* prefers to view concurrency in the context of day-to-day performance. Under this theory, if the first delay started on day one, and the second delay started on day two, they would not be concurrent – the delay associated with the first event would create float in the entire project so the second delay could not also be on the co-critical path. In the case where two independent delay events act on the same activity, the same rational applies: the first delay event causes the delay, while the second does not. *Literal* concurrency generally identifies fewer concurrent delays than *functional* concurrency. Since the *literal* concurrency requires the delay events to occur at the same time and *functional* concurrency only requires that the events occur within the same measurement period, it is very likely that more concurrency will be recognized under the *functional* theory. The *literal* theory requires the forensic analyst to look inside a monthly update. In one sense, this approach vitiates the analysis of monthly progress because the status depicted at the end of the month is insufficient.

The difference in outcome between the literal and functional theory is significant. Given the same network model, the literal theory practitioner will find less concurrency -- many more compensable delays for both parties. The functional theory practitioner will find many of those delays to be concurrent and hence excusable but, depending on the terms of the contract, non-compensable for both parties. It is also possible that the ultimate outcome may be similar when, under the literal theory, the compensation due one party is cancelled by the compensation due the other party. The only significant difference, despite the fact that the canceling effect (functional) operates under both theories, is the timing of the canceling effect and its impact on the damage calculation (literal).

Under the literal theory, an owner delay and a contractor delay of equal duration, occurring at different times are calculated as a period of compensable delay for the owner and a separate period of compensable delay of equal length for the contractor. The two periods will neither cancel each other out in time, nor money, since the contractor is likely to get a time extension for the owners delay and it is unlikely the owner's liquidated/stipulated damages rate will not be equal to the contractor's extended project rate. So, despite the apparent canceling effect, there is still potential of award of compensability to one side or the other. In contrast, under the functional theory, the canceling effect is realized *before* calculation of damages; hence there will be no offsetting calculation for damages.

The functional theory also recognizes the real-world limitations of exactly measuring delays and limitations of scheduling accuracy. While CPM schedules measure activities and events to the day, it is often difficult to retrospectively identify, with the exactitude of a day, the events on a project. By measuring possible concurrent delays with a measurement period larger than a day, the functional theory accommodates this real-world limitation. At the same time as the measurement period expands, it is likely that more delays will get treated like concurrent delays.

When evaluating the relevance of the time period, it is important to consider whether the concurrency analysis is being performed contemporaneously or forensically. Concurrent Delay analysis is frequently applied on projects that are still under construction because the full scope of the impact may not yet be known. Both parties to a construction contract often recognize that a full and final settlement of delay on a contemporaneous basis is not only compliant with the terms of the contract, but it provides a means to effectively balance risk on delays that are not yet complete. Contemporaneous analyses therefore, are often more *functional* than they are *literal.* When delay analyses are performed forensically, however, the standard-of-care increases because the settlement is likely to be based on technical proof rather than mid-project business decisions. Accordingly, forensic concurrency analyses are more likely to be *literal* in nature.

2. Least Float vs. Negative Float

The use of Negative Float or Longest Path Theory (Subsection 4.3.A.2.) for identification of critical activities can have a profound effect on the calculation of concurrent delay. The disparity stems from divergent approaches to criticality. Virtually all forensic delay methodologies provide for extensions of contract time on the critical path only. Therefore, the definition of the critical path is of utmost importance.

The Negative Float Theory assumes criticality on any activity that has negative total float relative to a contractual milestone. There is a certain practicality to this approach since most parties working from a CPM schedule will generally move to advance any activities that have negative total float because they are all essential to the maintenance or recovery of project delay.

The Longest Path Theory provides for criticality on the longest path only, even if other secondary paths are late with regard to a contractual milestone. Under the Longest Path Theory, all paths shorter than the longest path (even those with negative total float) have positive total float with respect to the longest path and are therefore not critical. In contrast, under the Negative Float Theory, any delays, occasioned by negative total float, occurring during the same measurement period are potential candidates for concurrency.

Concurrency analyses should always be consistent with the contract's definition of criticality. While it is beyond the scope of this document to catalogue the variations in contractual specifications, one relatively common definition is worth mentioning. Namely, some contracts include in the definition of concurrent delay that it cause a critical path delay. The requirement that the concurrent delay be critical, in effect, excludes other delay events with float values greater than the critical path from being evaluated for offsets against compensable delays. This view comports with the Literal Theory. Absent such contract definition, non-critical delays can be used to offset compensable delay on a day-for-day basis after the expenditure of relative float against the critical path. This view comports with the Functional Theory.

3. Cause of Delay vs. Effect of Delay

Another philosophical dichotomy that complicates the evaluation of concurrency is the difference between the proximate (immediate) cause of the delay and effect of the delay.

For example, assume a schedule activity with a planned duration of five days experiences work suspensions on the second day and the fifth day, thereby extending the duration by two days. The delaying events are on the second and the fifth day, but the delay-effect is on the sixth and the seventh day. The differences become much larger on activities with longer planned durations that experience extended delays. A good example would be delayed approval of a submittal that stretches for weeks and months.

The philosophical difference rests on the observation by the delay-effect adherents that there is no 'delay' until the planned duration has been exhausted. In contrast, the delay-cause adherents maintain that the identification of delay should be independent of planned or allowed duration, and instead should be driven by the nature of the event. The disadvantage of the delay-cause theory is that if there are no discrete events that cause a schedule activity to exceed its planned duration, it would have to fall back to the delay-effect method of identifying the delay. Conversely, in cases where the delay was a result of a series of discrete events, the delay-effect method of chronological placement of delay would often be at odds with contemporaneous documentation of such discrete events.

The difference in outcome is pronounced under the literal theory, since it affects whether or not a delay is identified as concurrent. Under the functional theory the significance to the outcome depends on whether the analyst is using a static method (MIP 3.1, 3.6 or 3.8) or a dynamic method (MIP 3.2, 3.3, 3.4, 3.5, 3.7 or 3.9). Using a static method, the cause-effect dichotomy makes no difference because the entire project is one networked continuum. But using a dynamic method, it does make a difference because the chronological difference between the cause and effect may determine the analysis interval in which the delay is analyzed.

There are two solutions to reconcile this potential dichotomy between the static and dynamic methods. One solution is to use the cause theory where discrete delay events are identifiable and to use the effect theory where there are no identifiable discrete events that led to the delay. But note that in many cases the identification of discrete causes is a function of diligence in factual research, which is in turn dictated by time and budget allowed for the analysis. The second solution is to review the delay on an activity basis and not to review the events on a daily basis within the event. This solution comports with the reality that delays that occur at the outset of an activity may be made up during the performance of that activity.

4. Frequency, Duration and of Analysis Intervals

Analysis interval refers to the individual time periods used in analyzing the schedule under the various dynamic methods (MIP 3.2, 3.3, 3.4, 3.5, 3.7 and 3.9). The frequency, duration and the placement of the analysis intervals are significant technical factors that influence the determination of concurrency. The significance of the analysis interval concept is also underscored by the fact that it creates the distinction in the taxonomy between the static versus the dynamic methods. The static method (MIP 3.1, 3.6, or 3.8) has just one analysis interval, namely the entire project, whereas the dynamic model segments the project into multiple analysis intervals.

a. Frequency & Duration

Concurrency is evaluated discretely for each analysis interval. That is, at the end of each period, accounting of concurrency is closed, and a new one opened for the next period. This is especially significant when analysis proceeds under the functional theory of concurrency in cases where two functionally concurrent delay events, one owner delay and the other a contractor delay, are separated into separate periods. If those delay events were contained in one period, they would be accounted together and offset each other. When they are separated, they would each become compensable to the owner and the contractor respectively. The analyst is recommended to analyze multiple-period events in both separate periods and combined periods to achieve the most accurate results.

However, the distinction between the functional and the literal theories does not disappear automatically with the use of multiple analysis intervals. Two delay events separated by time within one analysis interval will still be treated differently depending on which theory is used. The distinction becomes virtually irrelevant only when the duration of the analysis interval is reduced to a single day.

When multiple analysis intervals are used an additional dimension is added to the canceling effect that was discussed in the comparison of the literal theory to the functional theory. As stated above, the separation of two potential concurrent delay events into different analysis intervals causes the functional theory to behave like the literal theory. Because the change from one period to another closes analysis for that period and mandates the identification and quantification of excusable, compensable and non-excusable delays for that period, it is only after all the analysis intervals, covering the entire duration of the project, are evaluated that reliable results can be obtained by performing a ‘grand total’ calculation. In other words, the ultimate conclusion cannot be reached by selective evaluation of some, but not all, analysis intervals.

b. Chronological Placement

The general rule that all the intervals be evaluated will ensure the reliability of the net result. But the analyst can still influence the characterization of the delays by determining the chronological placement of the boundaries of the intervals, or the cut-off dates.

There are two main ways that the analysis intervals are placed. The first method is to adopt the update periods used during the project by using the data dates of the updates, which are usually monthly or some other regular periods dictated by reporting or payment requirements. The other is the event-based method in which the cut-off dates are determined by key project events such as the attainment of a project milestone, occurrence of a major delay event, change in the project critical path based on progress (or lack thereof), or a major revision of the schedule. Event-based cut-off dates may not necessarily coincide with any update period.

The most distinguishing feature of the event-based placement of cut-off dates is that there is significant independent judgment exercised by the forensic analyst in choosing that time period. Because the cut-off date is equivalent to the data date used for CPM calculation, it heavily influences the determination of criticality and float, and hence the identification and quantification of concurrent delays. Also, as stated above, the placement of cut-off date plays a major role in how the canceling effect operates.

5. Order of Insertion or Extraction in Stepped Implementation

In a stepped insertion (MIP 3.6, and 3.7) or extraction (MIP 3.8, and 3.9) implementation, the order of the insertion or extraction of the delay may affect the identity of potentially concurrent delays and their quantification.

As a general rule, for additive modeling methods where results are obtained by the forward pass calculation, the order of insertion should be from the earliest in time to the latest in time. For subtractive modeling methods the order is reversed so that the stepped extraction starts with the latest delay event and proceeds in reverse chronological order.

There are other systems, such as inserting delays in the order that the change orders were processed, or extracting delays grouped by subcontractors responsible for the delays. In all these seemingly logical schemes if chronological order of the delay events is ignored, the resulting float calculation for each step may not yield the data necessary for reliable determination of concurrent delays.

6. Hindsight vs. Blindsight

The difference between the prospective and the retrospective modes was addressed in Section 1. In this section however, we are reviewing two ways to view historic events in retrospective analysis. The first is “hindsight,” where the analysis uses all the facts, regardless of the contemporaneous knowledge, in determining what occurred in the past. The second is “blindsight” where the analysis evaluates events as-if standing at the contemporaneous point in time, with no knowledge of subsequent events. This RP deals primarily with the retrospective mode of analysis. The determination of concurrency made prospectively during the project is usually done using the functional theory so as to resolve potential concurrencies as they occur – essentially blindsight. However, such determinations may be discovered to be incorrect in hindsight using retrospective information. Thus, in the context of forensic schedule analysis, the analyst must be aware of the difference when reconciling the results of the retrospective analysis utilizing full hindsight with findings made during the project when the future was unknown.

The one place where this difference becomes technically relevant in the practice of forensic schedule analysis is in rectifying and reconstructing schedule updates (MIP 3.5 and 3.9). Specifically, the assignment of remaining duration to each partially progressed activity is highly dependent on whether the approach is hindsight or blindsight. Because CPM calculation of schedule updates depends, in part, on the value of remaining duration of activities at the data date, the difference in approach may affect the identification and quantification of concurrent delays.

The following figure illustrates the remaining duration of an activity using the blindsight method:

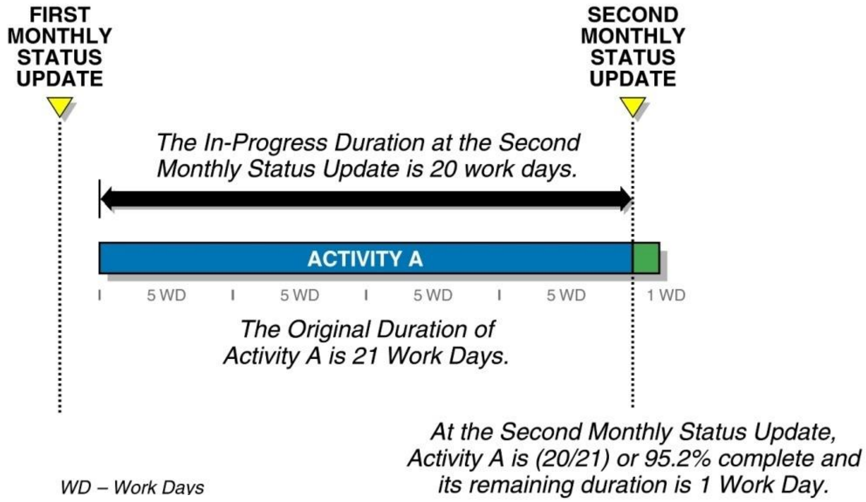


Figure 13 - Blindsight Method for Determining Remaining Durations of Activities in Progress

In the above example, Activity A has an original duration of 21 work days, starts several days after the first Monthly Status Update, and has been in progress 20 work days at the time of the second Monthly Status Update. Using the Blindsight method, and not knowing that any delay had occurred during the first 20 work days of progress, the remaining duration could be said to be only one work day at the time of the second Monthly Status Update. It would not be known until the activity was complete after the second Monthly Status Update that it's as-built duration was 25 work days.

The next figure below illustrates the remaining duration of an activity using the hindsight method:

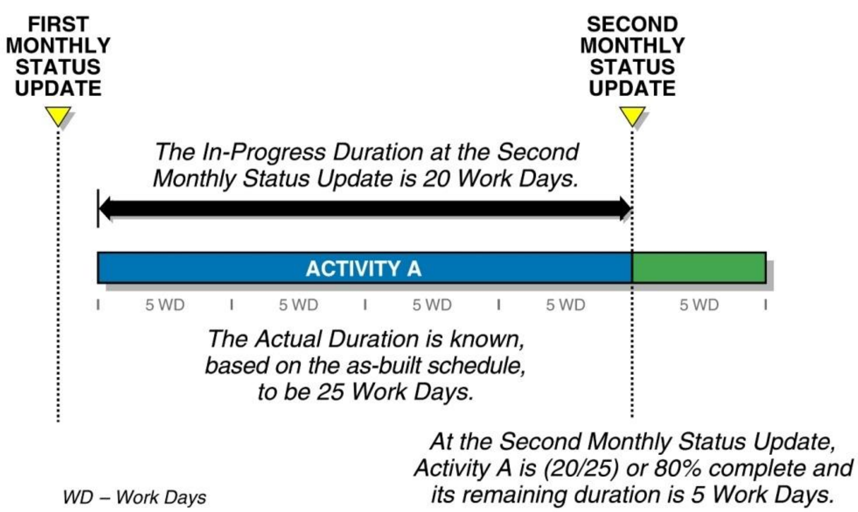


Figure 14 - Hindsight Method for Determining Remaining Durations of Activities in Progress

In this example, the same Activity A, which had an original duration of 21 work days, starts several days after the first Monthly Status Update, and based on the as-built data, finishes with an actual duration of 25 work days. The second Monthly Status Update occurs after 20 work days of progress on Activity A have occurred. Therefore, the analyst would conclude that the Activity A is 80 percent complete at the second Monthly Status Update, and would have a Remaining Duration of 5 work days at that time.

There is no prevailing practice, let alone agreement, on which practice ought to be used in the reconstruction of schedule updates. On one hand, the hindsight supporters maintain that it serves no purpose to ignore best available evidence and recreate updates, pretending that the as-built information does not exist. On the other hand, the blindsight supporters argue that the very purpose of reconstructing schedule updates is to replicate the state of mind of the project participants at the time of the update, because project decisions were made based on best available information at the time.

It is recommended that both approaches be evaluated in cases where difference in approach results in a significance variance.

E. Defining the Net Effect of Concurrent Combinations of Delay

If the contract documents are silent with regard to delay event definition, they are also likely to be silent on the net effect of concurrent combinations of delay. Under the foregoing delay definitions, there are just three potential combinations of discrete delay events. The following figure assumes the more common contractual situation where Force Majeure events are excusable but non-compensable events.

In the absence of specific contract language to the contrary, this Recommended Practice suggests the following protocol:

Delay Event 1		Delay Event 2	Net Effect
Force Majeure Delay [Time / No Compensation for Extended Overhead / No Liquidated/stipulated Damage Assessment]	concurrent with	**Contractor Caused Delay** [No Time / No Compensation for Extended Overhead / Liquidated/stipulated Damage Assessment]	**Excusable** [Time / No Compensation for Extended Overhead / No Liquidated/stipulated Damage Assessment]
Force Majeure Delay [Time / No Compensation for Extended Overhead / No Liquidated/stipulated Damage Assessment]	concurrent with	**Owner Caused Delay** [Time / Compensation for Extended Overhead]	**Excusable** [Time / No Compensation for Extended Overhead / No Liquidated/stipulated Damage Assessment]
Contractor Caused Delay [No Time / No Compensation for Extended Overhead / Liquidated/stipulated Damage Assessment]	concurrent with	**Owner Caused Delay** [Time / Compensation for Extended Overhead]	**Excusable** [Time / No Compensation for Extended Overhead / No Liquidated/stipulated Damage Assessment]

Figure No. 14 - Net Effect of Potential Concurrent Delay Combinations

Each of the foregoing conditions may result in an excusable, non-compensable delay (depending on the terms of the contract), which in turn typically results in at least four findings and remedies:

- Neither party benefits monetarily from the delay.
- The sole remedy for the delay is an extension of time.
- The right to compensation for either party is deemed offset by the compensation to the other party.
- The delay is treated as excusable and not within the control of either party.

F. Pacing

Pacing occurs when one of the independent delays is the result of a conscious and contemporaneous decision to pace progress against the other delay. The quality that distinguishes pacing from concurrent delay is the fact that pacing is a conscious choice by the performing party to proceed at a slower rate of work with the knowledge of the other contemporaneous delay, while concurrent delays occur independently of each other without a conscious decision to slow the work.

By pacing the work, the performing party is exercising its option to reallocate its resources in a more cost effective manner in response to the changes in the schedule caused by the other parent (non-pacing) delay and thereby mitigating or avoiding the cost associated with the resource demands. There may be no need to maintain the original schedule in the face of a known delay caused by the other party – no need to 'hurry up and wait'. In other words, it is the consumption of float created[13] in the pacing activity by the occurrence of the parent delay. Pacing delay is a real-life manifestation of the principle that work durations expand to fill the time available to perform them. It can take many forms. Work can be slowed down, resulting in extended work durations, or temporarily suspended, or performed on an intermittent basis. Whatever form it takes, the key is that it results from the performing party's reasoned decision to keep pace with another activity, which is called the parent delay, which is experiencing a delay.

There are two distinct circumstances to which the term, pacing delay, is often applied. The first circumstance, often referred to as direct pacing, occurs where the duration of a schedule activity is extended due to a delay in a predecessor activity on which the progress of the subject activity is directly dependent. An example would be the pacing of electrical conduit rough-in when the duration of metal stud installation is extended by delays. In such a case, because there is not enough work to sustain the continuous utilization of a full crew, the electrical subcontractor may order a crew size reduction, by temporarily reassigning some workers to other areas, slowing the progress. In either case it extends the overall duration of electrical rough-in. Although this is definitely pacing, it is <u>not</u> considered a pacing <u>delay</u> because the two activities are sequential and not concurrent.

The second type of pacing delay is where the paced activity has no direct dependency on the parent delay activity, sometimes called indirect pacing. The fact that it shares the same time frame is a function of schedule timing as opposed to construction logic. An example of this type of pacing would be the landscaping subcontractor who demobilizes its crew and returns at a later time because critical path work in the building has been delayed.

In this type of pacing, the sole relationship of the paced activity to the parent delay is the fact that the parent delay creates additional relative total float available for consumption by the paced

[13] The term 'creation' should not be interpreted to mean that total float is increased. In fact, the opposite is true. The parent delay adversely impacts the overall critical path of the project, thereby decreasing total float. What it creates (increases) is relative total float on the path of the paced activity relative to the total float on the path carrying the parent delay.

activity. The deceleration is achieved typically by reassignment or reduction of resources or entirely foregoing the procurement of resources that would have been otherwise necessary.

It should be clear that where the pacing defense is raised in answer to the identification of a potential concurrent delay, the pacing delay is not a distinct delay event but an alternate characterization or 'label' to describe and explain the concurrent delay event. Therefore, the pacing issue is relevant only to the extent that concurrency of delays is an issue. If there have been no potential concurrent delays identified, then pacing is irrelevant.

The term *pacing defense* is a misnomer, because paced performance, when properly undertaken, is a proactive rather than a reactive response to another party's parent delay. The use of the term *defense* implies that pacing is a forensic excuse rather than a contemporaneous option.

Pacing almost never occurs in the context of a literal method of concurrency analysis. Under the literal theory, the initial delay event would create float within the other near critical simultaneous activities. Since those activities had float relative to the new critical path, there would be no need to consider pacing.

Provided that pacing is not precluded by contract or local law, the contractor's right to pace its work in reaction to a critical path delay is a generally accepted concept. Thus, the contractor will not be penalized for pacing its work. This is consistent with the majority view that float, a shared commodity, is available for consumption on a 'first come first served' basis. Contracts that reserve float ownership to one party or the other may effectively preclude pacing as a management tool.

Pacing is irrelevant without the initial assertion of concurrent delay, and since concurrent delay is irrelevant where compensability is not at issue, the general acceptance of pacing strongly suggests that the contractor's right to pace would remove the owner's defense of concurrent delay and thereby make an otherwise non-compensable parent delay a compensable one. Alternatively, the owner can also pace performance. The owner's legitimate pacing would remove the contractor's defense of concurrent delay and thereby make an otherwise excusable contractor delay, non-excusable.

G. Demonstrating Pacing

In the absence of clear law or prevailing contractual language, the following criteria provide common sense guidelines for determining the legitimacy of pacing delays:

1. Existence of the Parent Delay

By definition, pacing delay cannot exist by itself. It exists only in reaction to another delay which is equally or more critical or is believed to be more critical than the paced activity. This calls for the calculation of relative total float between the parent delay and the pacing delay. Also, in cases where many different activities are being performed at the same time, it is unclear who is pacing whom. But one thing is clear: the parent delay must always precede the pacing delay. The existence of a parent delay is a mandatory requirement in legitimizing a pacing delay.

Quantitatively, the near-critical threshold can serve as a benchmark for the need to analyze for pacing delays, just like it serves to identify concurrent delays.

2. Showing of Contemporaneous Ability to Resume Normal Pace

Pacing is not realistic unless the party claiming it was pacing can show that it had the ability to resume progress at a normal, un-paced rate. Implicit in that party's ability to show that it could have completed the schedule activity on time if necessary is the fact that the party was able to reasonably determine or reliably approximate when the parent delay would end.

3. Evidence of Contemporaneous Intent

The case can be further strengthened by showing that the pacing was a conscious and deliberate decision that was made at the time of pacing. Without a notice signifying contemporaneous intent to pace, the claimant can use pacing as a hindsight excuse for concurrent delay by offering after-the-fact testimony. Typically, contemporaneous pacing notices are rare in any form, let alone specific written notices. Therefore this should not be a strict requirement of proof.

Paced performance is inherently risky because it is counter intuitive for any party to intentionally delay its performance on a project where time is of the essence. In order to mitigate such risk, it is always recommended that the party claiming the privilege provide the party responsible for the parent delay with notice of its intent to pace its performance. Unfortunately, such notices are exceedingly rare.

4.3. Critical Path and Float

A. Identifying the Critical Path

1. Critical Path: Longest Path School vs. Total Float Value School

In the early days of the development of the CPM, the longest path was the path with the lowest float. Using simple network logic (finish-to-start) only, the critical path of an un-progressed CPM network calculated using the longest path criterion or the lowest float value criterion is the same.

It is only when some advanced scheduling techniques are applied to the network model that the paths identified using these different criteria diverge (see Subsection 4.3.D.).

Most practitioners would agree that the longest path is the true critical path. Even with the use of advanced techniques, if basic network rules (see Subsection 2.1) are observed the total float value is a reasonably accurate way of identifying the critical path. But, note that float values are displayed using workday units defined by the underlying calendar assigned to the schedule activity instead of in 7-day calendar units. Therefore, activities on a chain with uniform network tension may display different float values.

2. Negative Float: Zero Float School vs. Lowest Float Value School

When a project is behind schedule, the network model may display negative values for float. Technically, this results from the fact that the earliest possible dates of performance for the activities are later than the latest dates by which they must be performed in order for the overall network to complete by a constrained finish date. Thus, the negative value is a direct indication of how many work days the schedule activity is behind schedule.

As discussed in Subsection 4.2.D.2.there are two schools of thought in interpreting the criticality of activity paths carrying negative float values. One school, which will be called the zero float school, maintains that <u>all</u> activities with negative float are, by definition, critical, assuming the definition of critical path is anything less than total float of one unit. The other school, which will be called the lowest value school, insists that only the activity paths that carry the lowest value are critical.

In the context of the two critical path schools, longest path versus total float value, the total float value adherents tend to align with the zero float thinking while the longest path adherents tend to think along the lines of the lowest float value school. However, neither one of these philosophical alignments is guaranteed, nor are they logically inconsistent.

Which one is correct depends on which principles are considered. If only CPM principles are used to evaluate the theories, the lowest value school is correct. The zero float school may have an arguable point if contractual considerations are brought into play, since all paths showing negative float are impacting (albeit not equally) the contractual completion date.

For the purpose of this RP, the procedures and methods use the lowest value theory as the valid criterion for criticality where negative float is shown. Thus, the true float value of a schedule activity carrying negative float will be calculated as the relative total float against the lowest float value in the network. For example, if the lowest float value in the network is minus 100, and another schedule activity shows a value of negative 20, the true float for that schedule activity, based on relative total float, is 80, assuming both activities are defined by the same calendar (see Subsection 4.3.D.2). The potential also exists for fragnets of activities to have lower total float than the project's longest or critical path. This occurs when activities are tied to intermediate project milestones (and not to overall project completion). If such a scenario is observed, the analysis should also consider the contractual relationship or requirement for the intermediate milestones.

B. Quantifying 'Near-Critical'

The purpose of quantifying the near-critical path is to reduce the effort of identifying and analyzing potential concurrent delays. A rational system of identifying all activities and delays that are near-critical is the first step in objectively streamlining the process of evaluating the schedule for concurrent delays. Thus, if the analyst chooses to analyze <u>all</u> delays and activities on a network, the quantification of near-critical is unnecessary. But in most cases, analyzing <u>all</u> activities, especially on large complex schedules, is excessively time consuming and unnecessary.

Near-critical delays have the greatest potential of becoming concurrent delays. This is because a near-critical delay, upon consumption of relative float against the critical path delay, will become critical. Therefore the near-critical delays are the most likely suspects of concurrency, and must be analyzed for partial concurrency to the extent that the net effect of that delay may exceed such relative float.

The determination of what a 'near critical' activity is depends on the following factors:

1. Duration of Discrete Delay Events

The insertion or extraction of delays affects the CPM calculations of a network model. Specifically, the duration of delays modeled in the analysis is directly proportional to the impact such delays have on the underlying network.

Because the effect results from insertion or extraction of delay, this is of obvious relevance to the modeled methods (MIP 3.6, 3.7, 3.8, and 3.9). But, it is also relevant to the dynamic observation methods where the underlying schedule updates were prepared during the project by inserting delay events.

The maximum duration of the set of all delay events would measure the greatest potential effect resulting from insertion or extraction. Averaging the duration of the set of all delay events would provide a less rigorous average measure. The maximum or the average measure is added to the value of the float value of the critical path to yield the near-critical threshold. Any schedule activity or path carrying a float value between that threshold and the value of the critical path is considered near-critical.

The practical effect is that the greater the duration of the delay events used in the model the greater the number of activities that must be considered near-critical and subjected to concurrency evaluation. Under this criterion, the most obvious way of minimizing the number of near-critical activities is to minimize the duration of the delay events. That is, a delay event of relatively long duration can be segmented into smaller sub-events for analysis and documentation.

While ensuring a finer granularity of delay events gives rise to added work in modeling and documenting those delay events, the trade-off is a lesser number of activities to analyze for concurrency.

2. Duration of Each Analysis Interval

The duration of the analysis interval is the length of time from the start of the segment of analysis to the end of that segment. In the dynamic methods (MIP 3.2, 3.3, 3.4, 3.5, 3.7, and 3.9) where the analysis is segmented into multiple analysis intervals, the measure would be the duration of each time period. In the static methods (MIP 3.1, 3.2, 3.6, and 3.8) [14] the duration of the analysis interval is the duration of the entire project or whatever segment of the project is represented by the schedule used for the analysis. Although this would mean that the static methods would have to perform a concurrency analysis on the entire network, it is both impractical and unnecessary to do so. Thus for methods that use the as-built as a component (MIP 3.1, 3.2, and 3.8), determination of near criticality can be made pursuant to the procedure established in Subsection 4.3.C below regarding the as-built critical path.

The concept underlying this criterion is the fact that the potential change in the critical path due to slippage, lack of progress or gain caused by progress during the analysis interval is equal to the duration of that interval. Thus, if the interval is one month, the maximum slippage that can occur, excluding non-progress revisions and delay insertions, is one month. Hence, near-criticality threshold would be set by adding 30 calendar days to the float value of the critical path.

This criterion is most relevant with the dynamic methods (MIP 3.2, 3.3, 3.4, 3.5, 3.7, and 3.9) that use the concept of analysis intervals. An implementation that uses large time periods would have to consider more activities near-critical than one that uses many small time periods. An extreme example of the latter is an as-planned versus as-built analysis that analyzes progress on a daily basis (MIP 3.2). This would have a near-critical threshold value of one day over the critical path.

The practical tradeoff is that by increasing the number of analysis intervals one can reduce the work load of concurrency analysis, and vice-versa.

[14] MIP 3.2 appears in both classifications because under some (but not all) implementations of MIP 3.2, the segmentation is merely a graphical tool for presenting a conclusion derived from a non-periodic analysis. Please refer to MIP 3.2 for details.

3. Historical Rate of Float Consumption

To augment the previous analysis interval criterion, the rate at which float is being consumed on a given activity-chain over time is worthy of consideration. The rate of consumption should be no more than the duration of the analysis interval per interval. Thus, where the interval is one month, if an activity chain is outside the near-critical threshold but is consuming more than 30 calendar days of float per month in the past updates[15], the trend indicates that it would become near-critical in the next period. Therefore, it should be considered near-critical even though it carries more relative float than the duration of the interval.

4. Amount of Time or Work Remaining on the Project

As the project approaches completion, CPM may not be the best tool to assess criticality. This is true especially in a problem project where many activities are being performed out-of-sequence in an attempt to meet an aggressive deadline. Even on a normal project, as the work transitions from final finishes to punch list work, CPM updates may be abandoned in favor of a list or matrix format of work scheduling. It is often said that near the end 'everything is critical'.

Reduced to an equation, the percentage of activities remaining on the network that should be considered near-critical is proportional to the degree of completion of the schedule.

Therefore, after 90 to 95 percent of the base scope and change order work are complete, the analyst may want to consider all activities on the schedule as near-critical regardless of float.

C. Identifying the As-Built Critical Path

As stated in Subsection 2.2, the as-built critical path cannot be directly computed using CPM logic since networked computations that generate float values can be generated only to the future (right) of the data date. Because of this technical reason, the critical set of as-built activities is often called the <u>controlling</u> activities as opposed to <u>critical</u> activities.

One method to <u>show</u> the as-built critical path is to create a collapsible as-built CPM schedule (Subsection 3.8.K.2) where the as-built schedule actual dates are converted into actual activity durations and actual driving lag durations. The total float values of the collapsible as-built schedule can be used to show the as-built critical path if the as-built logic was determined using the enhanced logic rules that not only uses the early-start and early-finish dates to simulate the as-built dates but also determine the proper late start and late finish dates. While there is acknowledgement that this is technically feasible, currently there is no agreement among practitioners on a common set of these enhanced logic rules.

The closest the analyst can come to determining the as-built critical path is to cumulatively collect from successive schedule updates the activities that reside on the critical path between the data date and the data date of the subsequent update. The same technique can be used to determine the as-built near-critical activities. If the updates are available, the following is the recommended protocol.

[15] Obviously this would be caused by reasons beyond just pure slippage. An example would be insertion of activities or a change to more restrictive logic.

a. Use all the critical and near-critical activities in the baseline schedule. If modifications were made to the baseline for analysis purposes, use both sets of critical activities, before and after the modification.

b. For each schedule update, use the critical and near-critical chains of activities starting immediately to the right of the data date.

c. Also use the predecessor activities to the left of the data date that precede the chains found in (b) above.

d. Use the longest path and near-longest path criteria in addition to the lowest float path criterion in identifying the activities.

e. If weather or other calendar factors are at issue, also use a baseline recalculated with an alternate calendar reflecting actual weather or other factors to gather critical and near-critical activities.

An enhanced protocol would add the following sets to the recommended protocol.

f. If appropriate, perform (b) through (d) above using different calculation modes[16] if they are available.

g. Where significant non-progress revisions were made during the updating process, repeat (b) through (d) using the progress-only, bifurcated schedules (see Subsection 2.3.D)

h. If appropriate, examine the resource-leveled critical path as opposed to hard-tied sequences, sometimes called preferential logic, based solely on resources.

i. Conversely, if resource constraint is at issue and the schedule logic does not reflect the constraint, insert resource-based logic to obtain a critical path that considers all significant constraints.

Objective identification of the controlling activities is difficult, if not impossible, without the benefit of any schedule updates or at least a baseline CPM schedule with logic. Therefore, in the absence of competent schedule updates, the analyst must err on the side of over-inclusion in selecting the controlling set of as-built activities. The determination must be a composite process based on multiple sources of project data including the subjective opinion of the percipient witnesses. All sources used to identify the as-built controlling path should be tabulated and evaluated for reliability. Contemporaneous perception of criticality by the project participants is just as important as the actual fact of criticality because important project execution decisions are often made based on perceptions. Perceived or subjective as-built critical paths can be based on:

- Interview of the hands-on field personnel.

- Interview of the project scheduler.

- Contemporaneous non-CPM documentation such as:

  - monthly update reports.

  - meeting minutes.

[16] For example, in *Primavera Project Planner*: retained logic and progress override modes.

- daily reports.

D. Critical Path Manipulation Techniques

There are various ways of creating, erasing, decreasing, inflating, or hiding float and manipulating the critical path of a CPM network.

These manipulation techniques can be used prospectively during the preparation of the baseline and the project updates as well as in the process of preparing the forensic models (MIP 3.6, 3.7, 3.8, and 3.9). This does not mean that the observational methods (MIP 3.1, 3.2, 3.3, 3.4, and 3.5) are immune from manipulation. Since they rely on the baseline and the updates, the source schedules must be checked for manipulation prior to use in the forensic process. Also, during the forensic process, the dynamic methods are subject to manipulation through the frequency, duration, and placement of analysis intervals (Subsection 4.3.B.2) and through subjective assignment of progress data in reconstructing updates (MIP 3.5).

The use of these techniques per se is not evidence of intentional manipulation. It must be stressed that there are legitimate uses and good reasons, albeit limited, for these features. Even in the absence of ‘good reason’, the feature could have resulted from laziness or even misguided attempts to improve the schedule. At any rate, schedules used for forensic schedule analysis must minimize the use of these techniques (see Subsection 2.1).

The policy of this RP is to be ‘software neutral’. This means that procedures and recommendations are made without regard to the brand or version of software used for analysis. However, the examples of techniques used to manipulate results, listed below, contain descriptions of the features found in some software manufacturer’s manuals

1. Resource Leveling & Smoothing

This technique uses available float to balance the resources necessary for executing the schedule. Some analysts maintain that resource leveling is the technical embodiment of pacing (see Subsection 4.2.F).

Resource leveling is the process of determining and minimizing the effect of resource availability on the schedule. Resource leveling can be used to resolve resource conflicts by rescheduling activities to times when sufficient resources are available. When resources are not available, activities can be split; activity durations can be stretched to reduce their resource per time period requirements; or, activity durations can be compressed to take advantage of ample resource supplies. During forward leveling, activities may be shifted to a later date (the leveled date). In backward leveling, activities may be moved earlier in time.

Resource smoothing is an optional resource leveling method that resolves resource conflicts by delaying activities that have positive float. Resource smoothing uses the available positive float and incrementally increases the availability limits.

2. Multiple Calendars

Float values are displayed using workday units defined in the underlying work-day calendar assigned to the activity instead of in calendar-day units. Therefore, activities in a logic sequence but with different calendars may display different float values.

All things being equal, activities using a more restrictive work-day calendar, such as one that excludes the winter months for work, carry less float than activities with less restrictive work-day calendar. Thus, by adding or removing a few holidays in the calendar, float can be manipulated.

While highly impractical, the only way to avoid gaps, discontinuities, and work-day conversions is to use only one calendar consisting of a seven-day week.

3. Precedence Logic / Lead & Lag

Simple logic is finish-to-start with a lag value of zero, denoted as FS0. Other known types of logic relationships are start-to-start (SS), finish-to-finish (FF), and start-to-finish (SF). Most software allows the use of these logic types along with the use of lead and lag values other than zero, including negative values. The use of lag values greater than zero with FS-type of logic absorbs otherwise available float. It is possible to assign lag values that are less than zero, called negative lags. Negative lags associated with the FS-type of logic have the effect of overlapping the associated schedule activities, thereby increasing float.

- ***Lag***: An offset or delay from an activity to its successor. Lag can be positive or negative; it is measured in the planning unit for the project and based on the calendar of the predecessor activity.
- ***Lead Time***: An overlap between tasks that have a dependency. For example, if a task can start when its predecessor is half finished, the analyst can specify a finish-to-start dependency with a lead time of 50 percent for the successor task. The analyst enters lead time as a negative lag value or as a percent complete lag value in some software packages.
- ***Lag Time***: A delay between tasks that have a dependency. For example, if the analyst needs a two-day delay between the finish of one task and the start of another, the analyst can establish a finish-to-start dependency and specify a two-day lag time. The analyst can enter lag time as a positive value.

4. Start & Finish Constraints

Setting a start constraint to a date that is later than what would be allowed by a controlling predecessor would decrease the float on the schedule activity. Similarly, setting a finish constraint to a date that is earlier than what would be allowed by a controlling predecessor would also decrease float on the schedule activity. Both techniques can be used to force activity paths to carry negative float.

There are also features that force the schedule activity to carry no total float or no free float. Also certain types of constraints force the assignment of zero float value by fixing dates on which the activity will be performed, overriding associated precedence logic.

5. Various Calculation Modes

Fundamental schedule and float calculation methods can usually be selected by the analyst, further complicating the effort to identify the critical path and quantify float. Below are examples related to various methods of schedule calculation, duration calculation, and float calculation.

a. Schedule Calculation

- ***Retained Logic:*** If the analyst selects retained logic, remaining activities are scheduled with out-of-sequence progress according to the network logic. When used, scheduling software schedules the remaining duration of an out-of-sequence activity according to current network logic - after its predecessors.

- ***Progress Override:*** Progress override ignores logic and affects the schedule only if out-of-sequence progress occurs. If the analyst selects progress override, remaining activities are scheduled with out-of-sequence progress as though they have no predecessors and can progress without delay. Not only does the successor activity act as if it no longer has any predecessor, the float of the predecessor activity also reflects the loss of that successor relationship. Progress override treats an activity with out-of-sequence progress as though it has no predecessor constraints; its remaining duration is scheduled to start immediately, rather than wait for the activities predecessors to complete.

b. Duration Calculation

- ***Contiguous Activity Duration:*** Contiguous activity duration requires that work on an activity occur without interruption. For early dates, this type of logic affects the start dates for an activity when the finish dates are delayed by a finish relationship from a preceding activity or by a finish constraint. If the finish dates of an activity are delayed, the start dates are delayed also.

- ***Interruptible Activity Duration:*** For early dates, interruptible scheduling affects how start dates of an activity are treated when the finish dates are delayed by a finish relationship from a preceding activity or by a finish constraint. If the finish dates of an activity are delayed, the start dates are not delayed. – The duration of the activity is stretched, allowing the work to be interrupted along the way.

6. Use of Data Date

- Reliable calculation of schedule updates requires the use of the concept of data date or status date is generally the starting point for schedule calculations. Generally, the data date is changed to the current date when the analyst records progress.

7. Judgment Calls during the Forensic Process

Any of the above techniques can be abused to effect discretionary decisions by the forensic analyst to influence the analysis in favor of the client. There are two instances in the forensic process that are especially sensitive to such influence because they directly affect the schedule variables at the data line. They are:

- Frequency, duration, and placement of analysis Intervals (see Subsection 4.2.A.3).

- Hindsight vs. blindsight update reconstruction (see Subsection 4.2.A.5).

E. Ownership of Float

In the absence of contrary contractual language, network float is a shared commodity between the owner and the contractor. Conventional interpretation of the principle of shared float allows the use of float on a first-come-first-serve basis, thereby allowing the owner to delay activities on that path up to the point where float is consumed. Therefore, as a corollary, if pacing is defined as the consumption of float, it follows that both owners and contractors are allowed to pace non-critical work.

Project float is the time between the last schedule activity on the baseline schedule and the contractual completion date where the contractual completion date is later than the scheduled completion date. In this case, in the absence of contrary contractual language, project float is owned solely by the contractor.

4.4. Delay Mitigation and Constructive Acceleration

A. Definitions

Acceleration: *A*ll or a portion of the contracted scope of work must be completed by the contractor earlier than currently scheduled. The accelerated work may be required as a result of: (a) direction of the owner or its agents (directed acceleration); (b) conduct of the owner or its agents without explicit direction (constructive acceleration); or (c) events within the responsibility of the contractor resulting in possible delay that the contractor decides to recover or mitigate. Acceleration typically has a cost associated with this performance.

Directed Acceleration: Formal instruction by the owner directing the contractor to: (1) complete all or a portion of the work earlier than currently scheduled; (2) undertake additional work; or, (3) perform other actions to complete all, or a portion, of the contract scope of work in the previously scheduled timeframe that otherwise would have been delayed. This could include mitigation efforts that usually have no costs associated with them.

Constructive Acceleration: (1) A contractor's acceleration efforts to maintain scheduled completion date(s) undertaken as a result of an owner's action or inaction and failure to make a specific direction to accelerate; **[4]** (2) Constructive acceleration generally occurs when five criteria are met: (a) the contractor is entitled to an excusable delay; (b) the contractor requests and establishes entitlement to a time extension; (c) the owner fails to grant a timely time extension; (d) the owner or its agent specifically orders or clearly implies completion within a shorter time period than is associated with the requested time extension; and, (e) the contractor provides notice to the owner or its agent that the contractor considers this action an acceleration order. **[4]** (3) Acceleration is said to have been constructive when the contractor claims a time extension but the owner denies the request and affirmatively requires completion within the original contract duration, and it is later determined that the contractor was entitled to the extension. The time extension can be for either additional work or delayed original work. **[5]** (4) Constructive acceleration occurs when the owner forces the contractor to complete all or a portion of its work ahead of a properly adjusted progress schedule. This may mean the contractor suffers an excusable delay, but is not granted a time extension for the delay. If ordered to complete performance within the originally specified completion period, the contractor is forced to complete the work in a shorter period either than required or to which it is entitled. Thus, the contractor is forced to accelerate the work. **[6]** (5) Acceleration following failure by the employer to recognize that the contractor has encountered employer delay for which it is entitled to an EOT (extension of time) and which failure required the contractor to accelerate its progress in order to complete the works by the prevailing contract completion date may be brought about by the employer's denial of a valid request for an EOT or by the employer's late granting of an EOT. This is not (currently) a recognized concept under English law. **[1]** (6) Constructive acceleration is caused by an owner failing to promptly grant a time extension for excusable delay and the contractor accelerating to avoid liquidated/stipulated damages. **[7]**

Disruption: (1) An interference (action or event) to the orderly progress of a project or activity(ies). Disruption has been described as the effect of change on unchanged work which manifests itself primarily as adverse labor productivity impacts. **[4]** (2) *Schedule disruption* is any unfavorable change to the schedule that may, but does not necessarily, involve delays to the critical path or delayed project completion. Disruption may include, but is not limited to, duration compression, out-of-sequence work, concurrent operations, stacking of trades, and other acceleration measures. **[8]**

Out-of-Sequence Progress: Significant work performed on an activity before it is scheduled to occur. In a conventional relationship, an activity that starts before its predecessor completes shows out-of-sequence progress. **[2]**

Delay Mitigation: A contractor's or owner's efforts to reduce the effect of delays already incurred or anticipated to occur to activities or groups of activities. Mitigation often includes revising the project's scope, budget, schedule, or quality, preferably without material impact on the project's objectives, in order to reduce possible delay. Mitigation usually has no or very minimal associated costs. **[4]**

Recovery Schedule: A special schedule showing special efforts planned to recover time lost for delays already incurred or anticipated to occur when compared to a previous schedule. Often a recovery schedule is a contract requirement when the projected finish date no longer indicates timely completion. **[4]** Recovery schedules are usually proposals that must be accepted by the owner prior to implementation.

B. General Considerations

1. Differences between Directed Acceleration, Constructive Acceleration, and Delay Mitigation.

In practice, there are subtle distinctions between directed acceleration, constructive acceleration, and delay mitigation. For example, directed acceleration cost implies additional expenditure or money for recovery of either incurred or projected delay, as well as efforts to complete early – all at the direction of the owner. The term constructive acceleration applies to expenditure of money for efforts to recover either incurred or projected delay caused by the owner and without specific direction to do so. Delay mitigation generally refers to no-cost recovery efforts for incurred or projected delay.

In the case of acceleration, constructive acceleration, and delay mitigation, affected activities are usually on the projected critical path; thus, the objective of most acceleration or mitigation is to recover from anticipated delay to project completion. However, acceleration, constructive acceleration, and mitigation can occur with regard to activities that are not on the critical path. For example, an owner might insist that a certain portion of the work be made available prior to the scheduled date for completion of that activity. The contractor may mitigate non-critical delay by resequencing a series of non-critical activities to increase the available float.

There are circumstances in which acceleration measures are used in an attempt to complete the project earlier than planned. Those circumstances are usually classified as: (a) directed acceleration where the owner directs such acceleration and usually pays for the associated additional cost; or (b) voluntary acceleration in which the contractor implements the plan on its own initiative in the hope of earning an early completion bonus. Contractor efforts undertaken during the course of the project to recover from its own delays to activities are generally not considered acceleration, even if the contractor incurs cost as a result.

The causative link between a delay event and cost associated with constructive acceleration is diagramed below. The root cause of the impact results in a construction delay or projects a construction delay. This, in turn, results in the contractor identifying that it needs a time extension and requesting a time extension. The owner denies the time extension request but the need for recovery from the delay remains. The contractor then undertakes acceleration measures that could include increased labor. Increased labor, without a time extension, can result in loss of productivity.

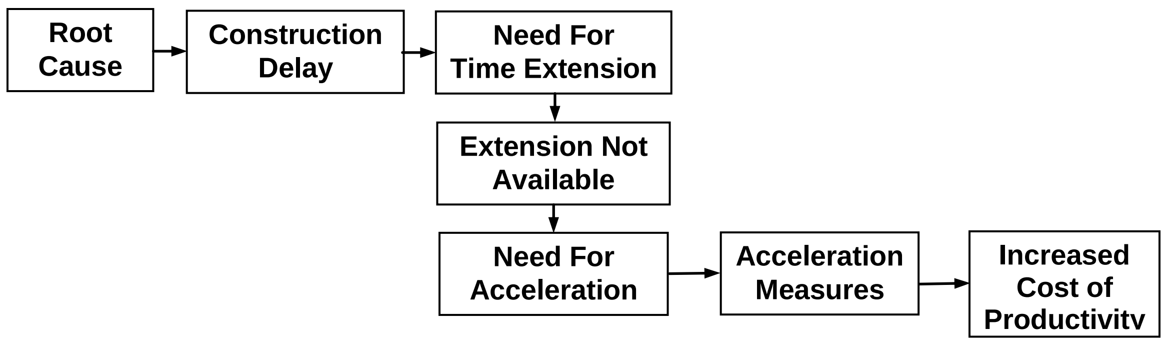


Figure 15 – Constructive Acceleration Flow Chart

A contractor's cost for acceleration, whether directed or constructive, is generally associated with the effort to engage more resources to perform the work during a unit of time than planned. These increased resources fall into the following major categories: (1) increased management resources; (2) increased equipment usage; (3) increased material supply; and (4) increased labor. The greatest cost associated with acceleration is usually increased labor. Since the amount of actual work remains unchanged in most acceleration efforts (assuming the planned scope of work has not increased), the increase in labor cost is a result of a decrease in labor productivity or the increase in the amount of overtime labor. Decreased labor productivity is caused by disruption to the planned sequence and pace of the labor. The greater the disruption to the work, the greater the inefficiency. Disruption can be the result of having more people working in the planned area during a specific time, or loss of productivity associated with individual workers working more hours than planned.

2. Acceleration and Compensability

Directed acceleration is always compensable to the contractor, although the parties may disagree on quantum. This is true regardless of whether the contractor is accelerating to overcome an owner-caused delay, or to recover from a *force majeure* event. Constructive acceleration follows this same pattern. If entitlement to constructive acceleration is established, the contractor may recover for a delay caused by the owner that the owner has refused to acknowledge and also for a *force majeure* event. This is different than the normal rule concerning damages associated with *force majeure* events. Typically, *force majeure* events entitle the contractor to time but no money. However, if an owner refuses to acknowledge a time extension for a force majeure event a contractor has no choice but to constructively accelerate so as to avoid the delay and possible liquidated/stipulated damages. As a result, the contactor is entitled to recover its cost associated with that constructive acceleration.

3. Delay Mitigation and Compensability

Delay mitigation is generally achieved through non-compensable efforts. These efforts are usually associated with changes in preferential logic so as to perform the work in a shorter timeframe. Mitigation applies to either incurred or predicted delays. There is no mitigation associated with efforts to complete early. Delay mitigation often has a small cost which is associated with the contractor's management of the schedule and the overall project. It is generally considered minimal and therefore ignored.

C. Elements of Constructive Acceleration

1. Contractor Entitlement to an Excusable Delay

The contractor must establish entitlement to an excusable delay. The delay can be caused by an action or inaction on the part of the owner that results in delay or it can be a *force majeure* event. In theory, a contractor can recover for constructive acceleration for work yet to be done. In this situation the owner takes some action that will result in the contractor expending acceleration costs to recover from the delay. The contractor could assert its entitlement even though the actual acceleration has yet to occur and the actual acceleration costs have yet to occur. In practice, since constructive acceleration occurs after the owner has denied a time extension, it is almost always resolved after the acceleration is complete and the contactor usually is arguing that it was actually accelerated.

2. Contractor Requests and Establishes Entitlement to a Time Extension

The contractor must ask for a time extension associated with the owner's action or the *force majeure* event. In that request, or associated with that request, the contractor must establish entitlement to a time extension. The owner must have the opportunity to review the contractor's request and act upon it. If the contractor fails to submit proof of entitlement to a time extension, the owner is able to argue that the opportunity was never given to properly decide between granting a time extension and ordering acceleration. The level of proof required to be submitted must in the end be sufficient to convince the eventual trier of fact that the contractor "established" entitlement.

In certain situations, it is possible that actions of the owner may negate the requirement for the contractor to request a time extension or to establish entitlement. In this situation, the theory is that the owner has made clear that a time extension will absolutely not be granted. Such cases are difficult to establish.

3. Owner Failure to Grant a Timely Time Extension

The owner must unreasonably fail to grant a time extension. This is closely related to the requirement that the contractor establish entitlement to a time extension. If the owner reasonably denies a request for time, as eventually decided by the trier of fact, then by definition the contractor has failed to prove entitlement. Therefore, the owner's decision not to grant a time extension where the contractor has shown entitlement must be unreasonable.

4. Implied Order by the Owner to Complete More Quickly

The owner must also, by implication or direction, require the contractor to accelerate. There are several different factual alternatives possible. First, a simple denial of a legitimate time extension, by implication, requires timely completion and thus acceleration. If this denial is timely given, the contractor can proceed. However, the best proof for the contractor is a statement or action by the owner that specifically orders the contactor meet a date that requires acceleration. Second, the owner could deny the time extension request and remind the contractor that it needs to complete on time. This is better than the first alternative above, but not as strong as the next alternative. Third, the owner could deny the time extension request and advise the contractor that any acceleration is the contractor's responsibility. This is probably the best proof for this aspect of constructive acceleration. All three of these options meet the test for an owner having constructively ordered acceleration. Examples of owner actions that meet this requirement include: (1) a letter from the owner informing the contractor that it must meet a completion date that is accelerated; (2) an owner demand for a schedule that recovers the delay; or (3) the owner threatening to access liquidated/stipulated damages unless the completion date is maintained. The fourth alternative arises when the owner is presented with a request for a time extension but fails to respond. The contractor is faced with either assuming that the time extension will be granted, or accelerating. Under this alternative, the owner's failure to timely decide, functions as a denial.

5. Contractor Notice of Acceleration

The contractor must provide notice of acceleration. As with any contract claim for damages, the owner must be provided notice of the claim. Even though the contractor has requested and supported the application for a time extension, the contractor must still notify the owner of its intent to accelerate or be actually experiencing ongoing acceleration. This is so that the owner can decide if it actually desired acceleration to occur, or, instead, the owner may decide to grant a time extension.

6. Proof of Damages

The contractor must establish its damages. For loss of productivity claims, the contractor is faced with developing convincing proof of decreased productivity. Actual acceleration is not required. A valid contractor effort to accelerate, supported by contemporaneous records, is sufficient to establish constructive acceleration. It is quite common that contractors accelerate to overcome delays but continue to be impacted and delayed by additional events and impacts that actually result in further delay to the project.

5. CHOOSING A METHOD

This part of the Recommended Practice discusses the choice of a forensic schedule analysis methodology. Because individuals generally work for one party to a dispute, there is often skepticism about the impartiality of the particular methodology chosen. Therefore, it is vitally important that all practitioners understand clearly what it takes to overcome this skepticism when choosing and using a particular delay evaluation method.

First, each claim is unique in that each deals with a different project, different contract documents, different legal jurisdictions, different dispute resolution mechanisms, and different fact patterns among other project execution factors. Likewise, each method discussed in this RP is different and each has certain technical factors to consider, including advantages and disadvantages. Because of the uniqueness and the need to consider multiple variables it is impossible to recommend one method that is the “best” method, or to rank the methods in order of preference.

Second, the selection of the analytical method should be based primarily on technical considerations related to the purpose, the timing, availability of data, and the nature and complexity of the delay and scheduling information.

Having selected the technically appropriate analysis method based on these criteria, the analyst must now consider the legal criteria, which varies from one jurisdiction to another. It is not possible nor is it the intent to list the selection guidelines of all the legal jurisdictions in this RP. The analyst is cautioned to seek the advice of legal professionals with specialized knowledge of the laws of the jurisdiction and forensic schedule analysis methods. This is true especially if the selection based on technical criteria must be reconciled with a different selection based on legal criteria.

Thirdly, there are a number of qualitative reasons, beyond technical schedule analysis reasons, that should be included in determining which forensic schedule analysis method is to be used for a particular claim. As in any commercial undertaking, while practical considerations are appropriate, these considerations must be secondary to the technical and legal considerations and should be used only when all appropriate technical and legal criteria have been met. Furthermore, the selection decision should be that of the analyst and not that of the client.

There is no requirement that the analyst select only one method to analyze a project. Some cases would necessitate the use of different methods for different phases of the project based on factors, including but not limited to, such as the nature of the claim (compensability versus excusability), types of delay causation, and source data availability.

This part of the RP discusses eleven factors that should be considered by the forensic schedule analyst when making a recommendation to the client and its legal counsel concerning this decision. Factors two, three, and five cover technical considerations. Factors one, nine and ten cover legal considerations. And factors four, six, seven, eight and eleven are practical considerations.

The forensic schedule analyst should consider each of these factors, reach a conclusion, and offer a recommendation with supporting rationale to the client and legal counsel in order to obtain agreement prior to proceeding with the work. Advance understanding of the analyst’s scope of work as well as the time, cost and resources required to perform the work should prevent surprise or disagreements during the drafting of the expert report or worse, at deposition.

5.1 Factor 1: Contractual Requirements

When a project is executed under a contract that specifies or mandates a specific schedule delay analysis method, then the choice of method is largely taken out of the hands of the forensic schedule analyst, and contract compliance is the prevailing factor. Some contracts, for example, now require that all requests for time extension (either during the life of the project or at the end of the job) be substantiated through the use of a prospective TIA (similar to MIP 3.6). As noted in this RP, several methods of forensic schedule analysis fall under this generic terminology. Most likely, the forensic schedule analyst will be required to use one of the additive modeling methods, either single base or multiple base, unless there are persuasive reasons why a different method would yield a more credible result. Care should be taken to ascertain whether the contract actually mandates the use of this analytical method in forensic situations (retrospective delay analysis) or whether it is intended solely for use in prospective delay analysis to aid in negotiation of time impacts due to changes or other delays. If the latter is the situation, then the choice of methodology could be made based upon factors other than contractual language.

On the other hand, if the contract documents are silent on which schedule delay analysis method is to be used when attempting to prove entitlement to a time extension or time related compensation, then the forensic schedule analyst is free to use any of the methods identified in this RP to support such requests. However, even when the contract is silent on methodology, contract language may still constrain the forensic schedule analyst's choice of methods. For example, some contracts contain language requiring that all time extension requests document that the event "…impacted the critical path of the project schedule" or "…caused or will cause the end date of the project schedule to be later than the current contract completion date." Thus, while this language does not dictate a schedule delay analysis method, it probably compels the forensic schedule analyst to use one of the observational dynamic, additive modeling, or subtractive modeling techniques. Also, it precludes the use of any method that does not identify or analyze a critical path such as a listing of delay events or a bar-chart analysis.

Thus, the first factor to be considered is the existence of an unambiguous contract requirement describing the documentation or method to be used to support requests for time extensions or time related compensation. Forensic schedule analysts should adhere to the requirements of the contract and to the applicable codes and laws under which the contract is governed. However, it is not uncommon that requirements set forth in contracts are unclear or ambiguous (such as a contractual reference to a "but-for TIA") or patently erroneous references such as contract language requiring the use of an "impacted as-built analysis". It is hoped that adoption and use of the terminology contained in this RP may help prevent such situations in the future. The forensic schedule analyst may want to use this RP as a mechanism to discuss the issue of differing forensic analysis methodologies with the client, legal counsel, and the other parties and help all focus on an appropriate method to be used.

5.2 Factor 2: Purpose of Analysis

Generally, the purpose of forensic schedule analysis is to quantify delay, determine causation, and assess responsibility and financial consequences for delay. Forensic schedule analysis studies how specific events impact a project schedule. Thus, the forensic schedule analyst uses contemporaneous project documentation to determine which events may have caused delay (including event identification, start and completion dates, activities impacted by the event, etc.). The forensic schedule analyst then applies or relates these events in some orderly manner to the schedules employed on the project. Once the events are added to, removed from, or otherwise identified in the schedule, a determination can be made concerning whether any or all of the events caused the project to complete later than planned. From this determination, assessment of causation and liability can be made based on the terms and conditions of the contract and the applicable law.

With respect to a particular project, the purpose of forensic schedule analysis is to determine if a party is entitled to time extensions or delay compensation as a result of certain events. Once the forensic schedule analyst has assessed the events that occurred on the project, then consideration should be given to issues such as concurrent delay, pacing delay, delay mitigation, etc. If the forensic schedule analyst, for example, needs to investigate whether concurrent delay is a major factor in the analysis of project delay, then the choice of method will be limited to those methods that specifically provide for concurrent delay identification and analysis. In such a situation, the forensic schedule analyst may be more likely to recommend one of the observational dynamic or modeled methods. If the purpose of the forensic schedule analysis is to demonstrate only excusable, non-compensable delay, numerous methods are available since the forensic schedule analyst will probably not need to deal with concurrent delay. If the purpose is to demonstrate compensable delay, other methods may be more appropriate. If the purpose of the analysis is to investigate the contractor's ability to complete work early in conjunction with a delayed early completion claim or how the timeframe available for the contractor to perform was compressed, again some schedule delay analysis methods may be better than others. Figure 16 below, generally summarizes the suitability of the nine MIP's for some typical forensic uses of CPM schedules.

Forensic Use of Analysis	METHOD								
	3.1	3.2	3.3	3.4	3.5	3.6	3.7	3.8	3.9
Non-Compensable Time Extension	OK	OK	OK	OK	OK	OK	OK	OK	OK
Compensable Delay	OK	OK	OK	OK	OK			OK	OK
Right to Finish Early Compensable Delay								OK	OK
Entitlement to Early Completion Bonus	OK	OK	OK	OK	OK	OK	OK	OK	OK
Disruption Without Project Delay	OK	OK	OK	OK	OK	OK	OK		
Constructive Acceleration				OK		OK	OK		

Figure 16 – Some Methods are Better Suited for Certain Purposes Than Others

5.3 Factor 3: Source Data Availability and Reliability

As discussed in this RP and emphasized heavily in the source validation protocols, the choice of a particular forensic scheduling methodology is substantially influenced by the availability of source data that can be validated and determined reliable for the purpose of the analysis. If, for example, the project records show that there exists only a baseline schedule but no schedule updates for the duration of the project, then the observational MIP's 3.3 and 3.4 cannot be utilized.

As a result, it is incumbent on the forensic schedule analyst to determine the amount of contemporaneous project documentation available and assess its quality. Then the forensic scheduler needs to review a sampling of the project documentation to determine if the data is reliable for the purpose of the delay analysis. Once these reviews have been completed, the forensic scheduler can formulate a plan for the forensic schedule analysis effort and make a recommendation concerning which forensic schedule analysis method can and should be employed on the claim. Figure 17 below shows the source schedules that are required to implement the minimum basic protocol for each MIP. Enhanced protocols would typically require additional schedule sources.

Source Schedules or Data	METHOD								
	3.1	3.2	3.3	3.4	3.5	3.6	3.7	3.8	3.9
Baseline Schedule	Min.	Min.				Min.	Min.		
Schedule Updates			Min.	Min.			Min.		Min.
As-Built Record	Min.	Min.			Min.			Min.	Min.

Figure 17 – Source Data Validation Needed for Various Methods

5.4 Factor 4: Size of the Dispute

One of the primary factors the forensic scheduler should keep in mind is the size of the dispute or the amount in controversy. In most situations, the choice of the forensic schedule analyst is constrained by how much a client has to spend to increase the probability of successful resolution of the dispute. This is most often determined by how much money is at stake. For example, if the delay damages being sought by the client are approximately US$100,000, then the forensic schedule analyst should recommend a relatively inexpensive forensic scheduling method that is still effective for its intended purpose. On the other hand, if the delay damages sought are US$50,000,000 then the range of methods to be considered is substantially expanded because of the greater scope and costs associated with analyzing a substantially larger claim. The forensic schedule analyst needs to recommend a forensic schedule analysis method that is both cost effective and suitable for the size of the dispute.

5.5 Factor 5: Complexity of the Dispute

When considering a forensic schedule analysis method, the forensic schedule analyst should do so with some knowledge of the complexity of the dispute in question and the number of events to be included in the forensic scheduling effort. For example, if the project in question is a linear project of relatively short duration, and only three specific delay events need to be considered, then a simple comparison of the baseline with the as-built schedule may be appropriate. On the other hand, if the project was a complex process facility, with a 5,000+ activity network, and a hundred or so discrete events occurring over a five year period, the forensic schedule analyst may need to recommend one of the observational or modeled methods that divides the project duration into smaller analysis periods to isolate and explain controlling delays. In this context, the forensic schedule analyst should also distinguish between the complexity of the dispute and the complexity of the forensic analysis. Some complex disputes can still be analyzed with a less complex analytical technique. And, some of the methods contained in this RP may not require analysis of every activity on the schedule but can be focused on the critical path and sub-critical paths or on key events and activities only, to reduce both the cost and the complexity of the analysis.

5.6 Factor 6: Budget for Forensic Schedule Analysis

Hand in glove with the size and the complexity of the dispute is the client's budget for the forensic schedule analysis. That is, what can the client afford to spend on forensic schedule analysis? The forensic schedule analyst needs to determine whether there are any budget constraints prior to making a recommendation on forensic schedule analysis methodology. The forensic schedule analyst should also keep in mind the overall cost of the various forensic scheduling methods when making a recommendation. For example, if the delay analysis method requires the testimony of ten or fifteen percipient witnesses in order to properly lay the groundwork for the analysis in arbitration or litigation, this cost too, should be taken into account.

If the law of the contract has a prevailing legal fees provision, then clients and their counsel may be willing to spend more on forensic schedule analysis than if the contract is under conditions commonly called the "American Rule" where each party pays their own cost, regardless of outcome. If the client is prepared to spend only a small amount for a forensic schedule analysis effort, then the forensic schedule analyst should consider using less expensive forensic scheduling methods or cost saving alternatives – such as using the client's in-house staff for certain tasks rather than outside consultant staff. Or, the forensic schedule analyst may find a method contained in this RP which is appropriate for the situation, but which does <u>not</u> require that all of the validation protocols be performed. If the forensic schedule analyst is required to take short cuts or rely upon the work of others to stay within a very tight budget, the forensic schedule analyst should advise the client and client's legal counsel of the potential risks of proceeding in this manner. The forensic analysis should keep in mind that if insufficient funding is available for the analysis that would be required to investigate and analyze the case, it may be proper and prudent for the analyst to refuse to undertake the assignment rather than knowingly use a methodology that is not appropriate.

5.7 Factor 7: Time Allowed for Forensic Schedule Analysis

There also may be occasions when the amount of time available to perform and produce a complete forensic schedule analysis is limited. Consideration should be given to the time required for research, data validation, and claim team coordination which may be extensive, as well as production of the report. If the contract contains a fast track arbitration clause which requires that hearings begin within ninety days of the filing of the arbitration demand, and all material to be used in the arbitration is to be exchanged with the other side no less than two weeks prior to the first hearing date, the forensic schedule analyst may be limited to a sixty day timeframe in which to perform the scope of work. In many situations, the need for forensic schedule analysis is not made early enough to allow complete flexibility in the choice of an analytical method or is made at the last minute due to time limitations designating testifying experts. In either situation, the forensic schedule analyst may have a very limited timeframe in which to complete its work. Should this be the case, then the forensic schedule analyst may be constrained to recommend short cuts or a method which can be completed in far less time than other forensic scheduling methods in order to meet the time available to perform the work. Again, the forensic schedule analyst should point out the risks of proceeding in this manner.

5.8 Factor 8: Expertise of the Forensic Schedule Analyst and Resources Available

If the forensic schedule analyst is experienced with only two or three of the methods identified in this RP and will be subject to challenge from the other side during *voir dire,* the forensic schedule analyst may be compelled to recommend use only of methods with which the analyst has experience. If the analyst determines that another method in which the analyst has little or no experience is more appropriate to the particular case then the analyst should be prepared to disclose that fact to the client. Additionally, if the forensic schedule analyst is to perform all analytical work individually with no assistance, the analyst may be constrained to recommend simpler methods which can be performed individually and will not require a staff of additional people processing data, making computer runs, etc.

5.9 Factor 9: Forum for Resolution and Audience

During initial discussions concerning the potential engagement, the forensic schedule analyst should seek advice from the client and its legal counsel on the most likely dispute resolution forum. What the forensic schedule analyst should seek is an opinion from those involved in the project, and their legal counsel, on whether the claim is likely to settle in negotiation, mediation, arbitration (and if so, under what rules), or litigation (and if so, in which court). If there is good reason to believe that all issues are likely to be settled at the bargaining table, or in mediation, then the range of options for forensic scheduling methods is wide open as the audience is only the people on the other side and they may be motivated, persuaded or willing to make decisions based upon a forensic schedule analysis method different than that specified in the contract. Almost any option which is objective, accurately executed and is persuasive is legitimately open for consideration. On the other hand, if legal counsel believes that the issue will end up in court or a government agency board, then the range of options available may be considerably narrowed because many courts and boards have adopted their own rules concerning forensic scheduling.

5.10 Factor 10: Legal or Procedural Requirements

Depending upon the forum for the dispute and the jurisdiction, the forensic schedule analyst must be aware of or ask about any contractual, legal, or procedural requirements that may impact the forensic analysis.

There may be other contractual, legal, or procedural rules impacting forensic scheduling that the forensic scheduling analyst should consider when making a recommendation concerning which forensic scheduling methodology to use on a particular claim. Consultation with the client’s legal counsel on these issues is essential.

5.11 Factor 11: Custom and Usage of Methods on the Project or the Case

The final factor to be considered is past history and methods. Typically, a forensic schedule analyst is not engaged until after preliminary negotiations have failed. Thus, the forensic schedule analyst needs to consider what delay analysis method was employed by the client and their staff earlier during the project, which was <u>not</u> acceptable to the other side in prior negotiations. Knowing this, the forensic schedule analyst generally should <u>not</u> recommend use of this technique, as it has already proven unsuccessful, unless the scheduler can determine that the client staff performed the method erroneously in their early efforts or that the basis of the previous ejection of the method was clearly erroneous. Additionally, the forensic scheduler should take into consideration the method that had been previously employed unsuccessfully, if known.

Not all of the above factors will be applicable to all delay claims, obviously. Nevertheless, a prudent forensic schedule analyst should consider each of the above factors, as well as any other relevant factors that emerge, to determine which apply to the claim at hand. Once these are identified, including their potential synergistic effect upon each other, the forensic schedule analyst should discuss each applicable factor with the client and their legal counsel prior to making a recommendation as to which method should be employed for the delay analysis. Failure to consider these factors could lead to substantial difficulties later on in claim settlement negotiations, arbitration, or litigation.

REFERENCES

1. *SCL Delay and Disruption Protocol*, Society of Construction Law, Oxon, United Kingdom, 2002
2. Primavera Project Planner – on-line software glossary
3. Microsoft Project – on-line software glossary
4. *AACE International Recommended Practice No 10S-90 "Cost Engineering Terminology"*, AACE International, Morgantown, WV, 2004
5. *Construction Delay Claims*, Third Edition, Barry B. Bramble, Esq., Michael T. Callahan, Esq., Aspen Publishers, New York, NY, 2006
6. *Construction Scheduling: Preparation, Liability & Claims*, Second Edition, Jon M. Wickwire, Esq., Thomas J. Driscoll, Stephen B. Hurlbut, Esq., Scott B. Hillman, Esq., Aspen Publishers, New York, NY, 2006
7. *How to Get Paid for Construction Changes: Preparation and Resolution Tools and Techniques*, Steven S. Pinnell, McGraw-Hill, New York, NY, 1998
8. *Proposed Specification Language Regarding Pacing*, Kenji P. Hoshino, AACE International Annual Meeting, 2006
9. *The American Heritage® Dictionary of the English Language, Fourth Edition*, Houghton Mifflin Company, Boston, MA, 2006

EDITORS **(2010 Revision)**

Kenji P. Hoshino, PSP CFCC
John C. Livengood, AIA, PSP CFCC
Christopher W. Carson, PSP

CONTRIBUTORS

(2010 Revision)

(June 23, 2009 Revision)

Kenji P. Hoshino, PSP CFCC (Author)
Andrew Avalon, PE PSP
Christopher W. Carson, PSP
Michael S. Dennis, CCC
Sidney J. Hymes, CFCC
John C. Livengood, AIA, PSP CFCC
Richard J. Long, PE
Mark F. Nagata, PSP
Jeffery L. Ottesen, PE PSP CFCC
Thomas F. Peters, PE CFCC
Dr. Anamaria I. Popescu, PE
Jose F. Ramirez, CCE
Mark C. Sanders, PE CCE PSP
L. Lee Schumacher, PE PSP
Stephen P. Warhoe, PE CCE CFCC
Ronald M. Winter, PSP
James G. Zack, Jr. CFCC

(June 25, 2007 Revision)

Kenji P. Hoshino, PSP CFCC (Author)
Robert B. Brown, PE
John J. Ciccarelli, PE CCE PSP
Gordon R. Costa, PSP CFCC
Michael S. Dennis, CCC
Edward E. Douglas, III CCC PSP
Philip J. Farrocco, PE
Sidney J. Hymes, CFCC
John C. Livengood, AIA, PSP, CFCC
Mark F. Nagata, PSP
Jeffery L. Ottesen, PE PSP CFCC
Thomas F. Peters, PE CFCC
Keith Pickavance
Dr. Anamaria I. Popescu, PE
Jose F. Ramirez, CCE
Mark C. Sanders, PE CCE PSP
Takuzo Sato
L. Lee Schumacher, PE PSP
Robert Seals, PSP
Ronald M. Winter, PSP
James G. Zack, Jr. CFCC

APPENDIX A: FIGURE 1 - NOMENCLATURE CORREPONDENCE FIGURE

Taxonomy	Level															
	1	RETROSPECTIVE														
	2	OBSERVATIONAL							MODELED							
	3	Static Logic			Dynamic Logic				Additive				Subtractive			
	4	3.1 Gross	3.2 Periodic		Contemporaneous Updates (3.3 As-Is or 3.4 Split)		3.5 Modified / Reconstructed Updates		3.6 Single Base[2]		3.7 Multi Base[1]		3.8 Single Simulation		3.9 Multi Simulation[1]	
	5		Fixed Periods	Variable Windows	All Periods	Grouped Periods	Fixed Periods	Variable Windows	Global Insertion	Stepped Insertion	Fixed Periods	Variable Windows or Grouped	Global Extraction	Stepped Extraction	Fixed Periods	Stepped Extraction
Common Names		As-Planned vs As-Built	Window Analysis		Contemporaneous Period Analysis, Time Impact Analysis, Window	Contemporaneous Period Analysis, Time Impact Analysis, Window Analysis	Contemporaneous Period Analysis, Time Impact Analysis	Window Analysis, Time Impact Analysis	Impacted As Planned, What-If	Time Impact Analysis, Impacted As-Planned	Time Impact Analysis	Window Analysis, Impacted As-Planned	Collapsed As-Built	Time Impact Analysis, Collapsed As-Built	Time Impact Analysis, Collapsed As-Built	Time Impact Analysis, Window Analysis, Collapsed As-Built

APPENDIX B: FIGURE 2 - TAXONOMY OF FORENSIC SCHEDULE ANALYSIS