

Article Simulation-Assisted Project Data Integration for Development and Analysis of As-Built Schedules

Muaz Fagiar ^{1,*}, Yasser Mohamed ^{2,*} and Simaan AbouRizk ²

- ¹ Senior Contracts and Claims Specialist, SMA Consulting, Edmonton, AB T5S 1G9, Canada
- ² Department of Civil & Environmental Engineering, University of Alberta, Edmonton, AB T6G 2W2, Canada; abourizk@ualberta.ca
- * Correspondence: fagiar@ualberta.ca (M.F.); yaly@ualberta.ca (Y.M.)

Abstract: As-built schedules are an essential tool for evaluating contractors' schedule performance and analyzing delay and lost productivity claims. Yet, most often construction schedules are not updated frequently and/or accurately as required, which limit the availability of as-built schedules. Furthermore, the retrospective development of as-built schedules, when sufficient and reliable project data is available, is a lengthy and costly process. This study describes a simulation-assisted modeling approach that automatically processes and integrates schedules progress data and develops as-built schedules at the activity level. The proposed method uses conceptual entities that are central to the operation of simulation models and whose content changes as they route through the schedule network model. The approach introduces (1) an entity information model that records relevant schedule information either in a materialized or virtual form, and (2) an entity lifecycle model that imitates the possible routes an entity instance may maneuver through in a schedule network model which, together, simultaneously respond to schedule logic and invoking duration changes. To demonstrate its effectiveness, a prototype based on the framework was developed using Excel and MS Project and was tested with a real case study. The study is expected to facilitate the development of as-built schedules for the analysis of delay and time extension claims.

Keywords: construction management; as-built schedules; construction simulation

1. Introduction

Contractors are usually required to submit baseline schedules that demonstrate their intended plan for delivering projects, as well as provide periodic progress updates during the execution phase. The updates are usually provided in different forms, such as daily site reports, monthly progress reports, meeting minutes, and/or updated schedules. This information becomes part of the project contract and is generally used for resolving relevant claims. Ultimately, this scattered information is gathered and transformed into as-built schedules, which represent the actual durations and execution sequences of activities following critical path method (CPM) principles. All project changes should be considered when developing such schedules, including, but not limited to, scope, schedule, and resource allocation changes [1–3]. As-built schedules are useful for many purposes, including verifying contractor progress on ongoing activities, project performance evaluation, claims analysis and validation, and as a reference when developing new project schedules [4]. However, developing and analyzing as-built schedules with dynamic CPM networks is difficult for many reasons, including inadequate site event documentation [2] and inaccurate schedule updates calculations [5], which limit the utilization of as-built schedules when analyzing delay claims.

To develop an as-built schedule, a collection of activity progress data over time is required. More importantly, if as-built schedules are to be used for delay analysis, a history of events that took place during the project execution phase must be documented and



Citation: Fagiar, M.; Mohamed, Y.; AbouRizk, S. Simulation-Assisted Project Data Integration for Development and Analysis of As-Built Schedules. *Buildings* **2023**, *13*, 974. https://doi.org/10.3390/ buildings13040974

Academic Editor: Irem Dikmen

Received: 10 March 2023 Revised: 31 March 2023 Accepted: 3 April 2023 Published: 6 April 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/).



readily available. Previous research focused on either (1) manually developing as-built schedules from unstructured project data [1,6] or (2) automating the as-built schedule development process [7–14]. The first approach is tedious and time consuming, because it requires a substantial effort to thoroughly review mass project documentation to sort and determine events that occurred during the project [15], and it often suffers from inaccuracies due to lost information [16,17]. Moreover, a standardized process for extracting meaningful data from scattered project data is lacking, consequently resulting in a prolonged process and augmented cost. A rule of thumb in claims preparation is that 70% of the time is spent on searching for and organizing information, with the analysis processes consuming the remaining 30% [18].

The second approach relies on the advancement of tracking technologies (digital cameras, laser scanners, and radiofrequency identification [RFID] sensors) [13], building information models (BIMs) and smart technologies (e.g., cyber-physical systems, big data, artificial intelligence, and smart robotics) [19]. Such technologies can be used to identify variations between actual and planned progress, so corrective measures can be taken for either schedule recovery or expenditure adjustment. Although this approach can improve the efficiency of collecting as-built data from construction sites, previous research was concentrated on progress monitoring and project controls, with as-built schedule development as a secondary focus. Additionally, adopting these new systems is an expensive process, because their application imposes new means of data collection and restricted project control tools.

This study reviews previous studies and highlights the need for a new as-built development approach. Additionally, it proposes and highlights the novel features of, a novel simulation-assisted approach to achieve (1) automated data processing, (2) integration of the impact of progress events on project schedules, and (3) assist in the development of as-built schedules. The architecture of the proposed simulation model includes two main constructs, an *entity information model* that uses attributes to document relevant event data and an *entity lifecycle model* that simulates the possible routes a simulation entity may take to maneuver through a schedule network in a way that responds to a predefined activity execution sequence and invokes changes to activity durations while maintaining the CPM principles. This entity-centric approach facilitates the development and analysis of as-built schedules in ways that current approaches do not. A real case study is used to demonstrate the merits of the proposed approach and validate its principles. The benefits of the newly proposed approach are also discussed, along with directions for future research. It is important to note that, while this study focuses primarily on as-built development and analysis, it is part of a larger research scope that aims to improve the overall analysis process by addressing the limitations of delay analysis techniques currently used in the construction industry [20].

2. Background and Literature Review

Much of the literature recognizes the value and importance of as-built records for the analysis of delay claims, as they can be used to validate as-built dates and evaluate the performance of contractors [4,7,21]. On the contrary, they can also be used to defend and validate time-related claims as they form the basis for most of the delay analysis methods [22].

In general, as-built schedules are developed either by creating an as-built schedule from scratch using progress records or by modifying a fully progressed schedule update as needed [22]. A subset of this approach is to create the as-built schedule by fully progressing a baseline schedule. Several studies have addressed the development of as-built schedules from contemporaneous and unstructured site records [1,6].

2.1. Developing As-Built Schedules from Unstructured Project Data

For instance, a study by Alkass et al. in 1995 introduced a computerized delay claim analysis (CDCA) system that integrates traditional scheduling tools with a delay expert

system. CDCA exports progressing activities into an editable format that are later updated and imported back into the scheduling tool and stored as a new schedule update to compare adjusted as-planned work with previous schedules [18]. A similar tool was later introduced by Al-Gahtani, Al-Sulaihi, and Iqupal (2016), which focused on implementing as-built versus as-planned delay analysis. The main challenges addressed by previous studies are the lengthy manual process and the need for additional resources to capture and sort the data [2]. Elzouni and Salem (2010) used neural-network pattern recognition (NN-PR) and statistical pattern recognition (S-PR) techniques, which map the CPM schedule, to classify planned progress patterns at a certain date, using the classification to measure and monitor actual progress patterns on the same day.

Other studies introduced new methodologies for determining a pre-structured set of data to automatically generate as-built schedules. Hegazy, Elbeltagi, and Zhang (2005) introduced an intelligent bar chart (IBC) that is made of spreadsheet cells in which activity durations are represented as groups of cells rather than bars to facilitate delay analysis. Each cell is then used to store the daily percentage complete for activities, general activity information, and other data relevant to delays. Navan and Haskaya (2006) developed a tool to generate monitoring and control information using information from a computerized daily site report (DSR) that stores as-built information. The tool uses this information to generate actual progress records that are later transformed to schedule updates using scheduling tools [23].

2.2. Developing As-Built Schedules Using Tracking Technologies

In another study, Hegazy and Abdel-Monem (2012) developed an email-based framework for progress tracking. The framework automatically sends daily emails to supervisors requesting as-built information for ongoing activities. Once the forms are returned, the system automatically reads the responses and updates the schedule with the recorded as-built information. The framework was later used to facilitate progress tracking and control of linear projects [24]. Further enhancements to the framework were later introduced by using cloud-based interactive voice response (IVR) technology to capture as-built information. Using the same principles, the system captures as-built records by either receiving update calls from supervisors or initiating automatic update calls for ongoing activities [25]. To automate the schedule updating process, Chin et al., 2008 proposed using RFID technology and 4D computer-aided design (CAD) models to monitor progress. In this approach, steal components are weighted and marked with RFID tags. Once steel components are installed and the tags are scanned, the progress of the relevant 4D CAD model is measured by comparing the total installed weight to the total planned weight. Notably, the application of RFID technology was limited by the manual tagging and scanning process.

To improve data collection efficiency, one study used digital images and AutoCAD to produce as-built schedules [16]. In principle, the study simulates 2D images or photos of the construction scene and compares them to a 3D CAD model to measure progress. Another study proposed superimposing site camera images on views of 3D models to recognize building structural components that were installed [9]. The study measures progress through user intervention by extracting site images and comparing them to a 3D model view to identify constructed components. Another study used a fixed position camera to take a sequence of images at different times, with the differences between consecutive images analyzed and compared to the 3D as-planned model [11]. A similar approach, also based on recognizing the difference between images, was used by Zhang et al., 2009.

Golparvar-Fard et al., 2009 proposed generating a 3D point cloud from a large number of daily site images and aligning and comparing the point cloud to an as-planned 4D model to identify constructed components. In another study, researchers used a Bayesian probabilistic model to measure the physical progress of structural components along with a continuation of research on calculating differences between planned and actual progress [26]. However, as these studies failed in producing as-built schedules, they are not useful for delay analysis. This failure is mostly due to the fact that the project schedules are defined at a different level of detail than the building components. To overcome this problem, Turkan et al., 2012, proposed an approach that integrates 3D point clouds, 4D CAD models, and 3D sensing technologies to automate progress tracking and schedule updating. In this approach, the coordinates of the 4D CAD model are compared with site condition data that are captured using a 3D laser scan to extract as-built objects, measure progress, and automatically update the project schedule [13]. However, the dates of the updated schedule were found to be incompatible with traditional scheduling software such as Microsoft Project or Oracle Primavera. Following the same principles, Son, Kim, and Cho (2017) used the 3D registration method proposed by Kim Son and Kim (2013), which employs a laser scanner to obtain a 3D point cloud that is later compared with a 4D BIM to identify the constructed components. The Kim Son and Kim (2013) study also proposed an as-built status revision process to modify inaccuracies caused by the incompleteness of the 3D point-cloud data set. As BIM is highly praised for its potential for efficient buildings management, some researchers have focused on the automation of Scan-to-BIM using occluded point clouds [27]. In a recent study, Zhu, Hwang, Ngo, and Tan (2022) proposed using a two-stage 3D object-detection approach using region-based convolutional neural networks (R-CNN) to reconstruct 3D models of non-Manhattan and multiroom buildings. However, the study also did not address the challenge of developing dynamic as-built construction schedules.

Although prior studies primarily focused on developing as-built schedules from existing data or automating data processing, an overwhelming majority of the proposed solutions focused on the prospective development of as-built schedules to serve the project control aspects of a project. Subsequently, prior studies failed to recognize the need to automate the retrospective development of as-built schedules. Only one study, by Shrestha and Jeong (2017), focused on automating the development of as-built schedules for highway projects from previously collected field data. The data were collected using a Daily Work Report (DWR) system that encompassed a variety of activities, such as ongoing construction activities, equipment types, labour and equipment hours, and quantity of work performed [28]. The proposed framework filters the data of a certain project, processes activity duration and quantity of the work performed, evaluates contractor performance, and develops a visual as-built schedule accordingly [29]. However, the schedule produced has serious limitations, including a restricted application area, a lack of CPM functionality, and a lack of planned schedule consideration.

3. Study Rationale

The current state of the art practice, represented by the previously discussed studies on as-built schedule development, has a number of gaps in knowledge that hinder the efficient utilization of as-built schedules for delay claims analysis. The reviewed methods do not meet the increasingly tight timelines for delay claims and the need to integrate heterogeneous project data sources and schedule updates to develop an as-built schedule while complying with CPM principles. Specifically, this study addresses the following three main gaps which include (1) a need for a practical approach for automated and retrospective development of CPM based as-built schedules, (2) a relatively long as-built schedule development time compared to the short decision-making window that governs most construction contracts for dispute resolution, and (3) a lack of structured approach for progress documentation and demonstration of controlling activities of completed work. Table 1 summarizes the capabilities of existing research and indicates the identified gaps in each of the approaches (highlighted in a double-line border).

										As-Built	Features			
	Main Research Focus					Basic			Enhanced/Advanced					
Study	edule Development	s Monitoring	ata Records	ayment	umunication	y Analysis	uilt Dates	unctionality	Auton	nation	nty Modeling	Traceability	vities Demonstration	Structure
	As-Built Sch	Progres	Site-D	P.	Com	Dela	As-B	CPM F	Prospective	Retrospective	Uncertai	Events	Controlling Acti	Data
[18]	√.					√	√ .	\checkmark						\checkmark
[1] [2] [16] [23]	\checkmark	\checkmark	\checkmark			\checkmark	$\begin{array}{c} \checkmark \\ \checkmark \\ \checkmark \\ \checkmark \\ \checkmark \\ \checkmark \end{array}$	\checkmark	\checkmark			\checkmark		\checkmark
[8] [9] [12] [30]		\checkmark		\checkmark	\checkmark	\checkmark	\checkmark	\checkmark \checkmark	\checkmark					
[17] [6] [13] [14]	\checkmark	\checkmark	\checkmark		\checkmark		\checkmark	√ √	\checkmark					
[25] [24] [29] [31]	\checkmark	\checkmark	\checkmark		\checkmark		\checkmark	\checkmark	\checkmark	\checkmark				\checkmark
[19] Proposed Simulation	V	V	√ √			\checkmark	V	\checkmark	√	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark

Table 1. Summary of previous studies.

4. Simulation Architecture

This study expands on the simulation-based framework that was introduced by Fagiar, Mohamed, and AbouRizk [20] for delay analysis. The framework takes advantage of simulation technology to model the dynamic of schedule changes as a project progresses by modeling and integrating activities changes such as durations, logic, status, start and finish dates, that are caused by unplanned events. While the framework includes delay analysis functionalities such as delay analysis and quantification algorithms, this study focuses primarily on the simulation aspects of the framework by describing the simulation architecture relevant to the development and analysis of as-built schedules.

The simulation is based on the concept of time-step analysis where the analysis result of a time-step forms the basis for the successor time-step. It models the dynamics of schedule changes by extracting relative activities and their logic based on a predefined time-step of one entity, integrating the impact of events relevant to the entity instance into the CPM computation, and lastly analyzing the data to provide results. To enable this process, conceptual entities that represent time units such as minutes, hours, or weeks, are generated. It is important to note the distinction between an entity and an event. While an entity is an object with a unit of time, an event is an instantaneous incident that changes a state of the schedule and/or influences the occurrence of other events. The simulation architecture is illustrated in Figure 1. Detailed descriptions of the proposed components are discussed as follows.



Figure 1. High-level Simulation Architecture.

4.1. Provisions

The proposed approach relies on data from three sources, which serve as inputs: the baseline schedule, schedule updates, and the schedule of events (SoE). The proposed approach utilizes the baseline schedule as an initial reference to measure time extension or acceleration, to allocate liability, and to support the as-built versus as-planned analysis. It is also used to model the planned sequence of activities and to estimate activity duration, logical constraints, and project calendars to calculate the start and finish dates of activities as per CPM computations. The proposed approach also supports available schedule updates to reflect schedule changes. It is important that schedules used in the analysis adhere to the best scheduling practices.

The SoE is a documentation of project events that may impact the project schedule, including, but not limited to, daily site events, management decisions, change orders, and others. The proposed approach requires certain attributes of an event to be documented in a certain format, which include event ID, description, cause, quantification type (fixed, uncertain, or formula), start time, parameters, responsible part, impact type (global or task specific), as well as references and their issue dates. The reader is referred to Fagiar, Mohamed, and AbouRizk [20] for detailed descriptions of these attributes and their formats. Transforming these inputs into the acceptable specification is based on a combination of actions from the analyst, along with automated verification and synthesis algorithms.

4.2. Simulation

Construction schedules are constantly subject to changes, depending on internal and external factors, such as planned activities, resource availability, approvals, and weather conditions. As such, a time-step simulation approach is most suitable to model the dynamics of CPM networks. It is reasonable that simulation entities may represent time units (hours, days, weeks), based on the desired level of abstraction. This poses certain modeling requirements, including (1) that each entity must be distinct and represent unique and 'Active' event(s) and (2) the capability to route through 'Active' activities, so that duration adjustments can be made where applicable. The 'Active' status of an event or an activity refers to the occurrence of the event or activity at that entity instance. These requirements

will be addressed in the following sections through a detailed explanation of the simulation components. The entity movement is facilitated through two main constructs, (1) an *entity information model* that contains relevant information of events and schedule activities and (2) an *entity lifecycle model* that defines possible routes through which an entity can proceed through a schedule network.

4.3. Entity Information Model

Simulation entities are initiated equally by default. To make them representative of distinct time points, certain key events and activity information must be carried by the entities. This often refers to descriptive attributes such as event ID, cause, and impact. To enable entities to carry both event and activity-related information, an integrated entity information model (EIM) is designed. Figure 2 illustrates the components of EIM, which include (1) simulation run-time data, (2) impacting events data, and (3) other project control data. The EIM is a record type where each of the fields is either a record of an attribute or collection of attributes.





As sown in Figure 2, the EIM is broken into three categories. The *Simulation time attributes* hold information about the simulation time of an entity instance, including the most current simulation timestamp as per the entity instance, cycle number, and the entity instance label (working or non-working day). The event attributes hold information about the occurrence of impacting events relevant to the entity instance, including event ID, cause, description, impact value (derived), impact type, and applicable impacted activities ID. An impact of an event on a project schedule can be expressed in three different ways, which are certain (known event duration), uncertain (uncertain event duration), and formula for events with impacts that can only be expressed formulaically. Events with uncertain impact are modeled using probability distributions such as uniform, triangular, or beta distributions. Depending on the type of events impact, the end date of each event is calculated.

A given event occurrence may be relevant to multiple entity instances; in such cases, event attributes will be recorded for all applicable entities. This means the event span is larger than an entity instance. Nesting events is supported and executed in parallel. The simulation allows both events that occur at specific times as well as periodic events to be modeled. When there are multiple events relevant to an entity instance, EIM will contain records of all events, route through all Active activities, and, accordingly, trigger change to their duration. The project control attributes are not currently implemented in this study; however, they are included to demonstrate how the EIM can be expanded to hold other information such as resources, cost, and risks.

4.4. Entity Lifecycle Model

Figure 3 shows the topmost levels, referred to as stages, of the entity lifecycle. The entity lifecycle model specifies the activities that an entity can be involved in as the entity routes through its lifecycle. It is specified using stages, where each stage consists of one or more guards. A guard is a condition (possibly a triggering event) that, when activated, enables entities to route through the network model and trigger changes in the activity durations. The guard's conditions range over the information model of the entity instance and are expressed using if/then rules. When these rules are evaluated to be TRUE (satisfying the condition under consideration), the guards become OPEN, allowing a set of associated actions (changes to activity durations) and other rules to be executed. For instance, if there was an Active Event at an entity instance, the Events' Schedule guard would become OPEN, allowing the entity to pass through to collect records of that Active event.



Figure 3. Entity life cycle model.

4.5. Run-Time Engine (RTE)

The run-time engine (RTE) is responsible for controlling the simulation execution; creating the simulation environment; initiating entities with a timestamp, cycle number, and label—and advancing the simulation time. At time zero ($\tau_0 = 0$), the starting simulation date is initiated as the smallest early start time in the project schedule. In other words, the simulation sets the starting simulation date as the earliest start time ([ES] _i) of the first activity in the project. The simulated start date will then be incremented by one entity throughout the simulation of the entire schedule. Hu (2013) argued that using large time-steps results in fast but inaccurate/unstable simulations. Therefore, it is recommended to use small entity span to achieve more precise results.

4.6. Schedule of Events Model

An event can be broken into smaller events (events slicing), which are referred to as Ebites in this framework to reduce the modeling complexity of an event, thereby facilitating integration into the project activities. Event slicing transforms events into E-bites by using the entity span as slicing criteria to identify E-bites relevant to an entity instance. All E-bites inherit the attributes of their parent events. The generation of E-bites begins by defining the slicing criteria, which are the time spans represented by the entity instance. Each E-bite is tagged with a timestamp and includes a Boolean attribute, which holds its status. This is initialized at CLOSE and becomes ACTIVE once its guard becomes OPEN. The guard status changes as per the status of its parent event. For example, if an entity was released and a parent event was active, the guard becomes OPEN. This means the entity will route through the event's schedule, and its information model will be updated as per the attributes of the parent event.

4.7. Network Model

Initial values of the activities' attributes (ID, name, duration) embedded at compile time (τ_0) are used to model the network of the project schedule. The proposed approach relies on basic CPM principles, including modeling precedence relationships, leads and lags, calendars, as well as both forward and backward passes computation through the schedule network. As schedule updates become available, they are used to update the

activities' attributes and execution logic. Modeling the aforementioned principles can be done programmatically in variety of ways, depending on preference. The modeled CPM is executed at the initiation to establish the planned project timelines. The framework supports continuous critical path identification and calculation at every time-step. It is important to note that executing the CPM at the initialization is also a vital step in verifying the modeled network.

A similar approach to event slicing is used for scheduling activities, where the durations of activities are broken down into smaller activities, referred to as A-bites. Each A-bite represents an entity span; its status attributes change as per the release of a matching entity. The Network model, however, has a guard for the network and guard for each activity. The network guard becomes OPEN once the status of a parent activity becomes ACTIVE, which allows the entity instance to route though the Network model. The activity guards become OPEN only when the network guard is OPEN, and the status of the A-bite becomes ACTIVE. In such a case, the entity routes through the parent activity and triggers changes to its duration as per the attributes of the relevant E-bites.

4.8. Data Integration

As shown in Figure 4, the integration process begins by advancing the simulation time from day zero (τ_0) to the starting simulation date (τ_1). The RTE then releases the first entity instance tagged with working or non-working labels as per the project calendar. When an entity instance is released, it may result in a series of changes in the status of parent events, E-bites, parent activities, and A-bites becoming ACTIVE. Guards become OPEN or CLOSED.

When τ_1 is a non-working day, the entity routes to the compiler without triggering any changes in the CPM network model; otherwise, the simulation explores the event's schedule to determine if there is any ACTIVE event relevant to the entity instance through an event-to-event sequence. In such a case, the event's guard becomes OPEN, allowing the entity instance to route through the event's schedule. When there is an ACTIVE E-bite, the simulation updates the EIM; otherwise, the entity simply routes to the compiler. For ACTIVE E-bites with global impact type, the impacted activities field in the EIM will be updated to include all the ACTIVE A-bites. Additionally, for ACTIVE E-bites with a task-specific impact type, the impacted activities field in the EIM will be updated to include the ID of the impacted activities regardless of their status.

The entity instance then transfers to the network model stage. When there is an ACTIVE parent activity, the network guard becomes OPEN, allowing the entity instance to route through the network model. The activity guards become OPEN when there are ACTIVE A-bites, allowing the entity to trigger changes to the duration of parent activities as per the impact field of EIM. Durations of all Active parent activities are updated simultaneously, and the CPM model will be executed accordingly. Thereafter, the entity routes through the compiler, and the simulation advances to the next time step (TS) by following the same updating process. Thus, it shifts from one entity to the next until it passes the timestamp of the last ACTIVE E-bite. If, at any time during the simulation, a guard has a CLOSED status, the entity routes to the compiler without triggering any changes in the network model. It is important to note that an activity's duration is only updated to a maximum of one entity span (a time-step). The update could either be positive or negative.



Figure 4. Data integration process.

4.9. Outputs/Compiler

The last stage in the entity lifecycle is the compiler responsible for capturing the information recorded in each EIM, for compiling the results of the CPM, and for identifying the controlling activities at every interval. Once the simulation is complete, the compiler uses this information to generate the as-built schedule, which is a result of incorporating all of the impacting events into the durations of the base schedule's activities. The planned and simulated timelines for each activity are recorded along with the history of the evolved events relevant to the activity. Then, a comparison of the as-built records and planned schedule of controlling and critical activities is visualized in a graphical and statistical fashion.

5. Case Study

5.1. Background

To verify the functionality of the framework, the prototype model was applied to a number of hypothetical case studies with various scheduling scenarios. Initial results demonstrated that the framework has several benefits due to the added features for asbuilt schedules. To demonstrate the observed benefits, the prototype was applied to a segment of a real-life tunneling project. The segment includes the excavation of a 162-m tunnel using a tunnel boring machine (TBM) and construction of a removal shaft. This segment was expected to be completed over 44 working days (59 days including weekends). The breakdown of activities involved in this segment and their estimated durations were defined in Microsoft (MS) Project, as shown in Figure 5.

						2018	May					2018	June		
Task Name 👻	Duration 👻	Start 👻	Finish 👻	22	27	02	07	12	17	22	27	01	06	11	16
₄ TBM tunnel	42 days	Mon 18-04-23	Tue 18-06-19												
Sandstone 68 m, 5.2 m/day	13 days	Mon 18-04-23	Wed 18-05-09				h								
Sandstone curve 47 m, 2.8 m/day	17 days	Thu 18-05-10	Fri 18-06-01				T T					Ы			
Sandstone 62.09 m 5.2 m/day	12 days	Mon 18-06-04	Tue 18-06-19									+			
4 Removal Shaft	31 days	Thu 18-05-10	Thu 18-06-21				r								
Mobilize to site	10 days	Thu 18-05-10	Wed 18-05-23				+			-h					
Prepare shaft liner	4 days	Thu 18-05-24	Tue 18-05-29		i i i i i i i i i i i i i i i i i i i										
Excavate shaft (4.8 m)	4 days	Wed 18-05-30	Mon 18-06-04								+	_ 1			
Continuously drill the shaft (20 m)	8 days	Tue 18-06-05	Thu 18-06-14									+		h	
Build safety wall and ladder	4 days	Fri 18-06-15	Wed 18-06-20											+	
Complete shaft construction	1 day	Thu 18-06-21	Thu 18-06-21												1



Another project schedule, as shown in Figure 6, was later provided by the contractor to accommodate a TBM alignment check that took 25 days and was accepted by the owner as contemporaneous evidence providing as-built information for evaluating schedule delay and changes in the critical path.



Figure 6. Updated schedule of the tunneling segment.

5.2. Automated Development of As-Built Schedule

This study relied on daily progress reports and monthly progress reports that were available from the project. These contemporaneous records were organized and sorted as per the format and attributes of SoE. This process helps with organizing scattered project information, and, subsequently, enables the automatic development of as-built schedules at any time-prospectively and/or retrospectively. A concept proving prototype was developed to support scenario experiments and implementations. The prototype model uses Excel for the event schedule and MS Project for the planned schedule, as well as front-end data processing through Visual C#.Net. MS Project data was manipulated and transferred automatically to the simulation using an MPJX library. Then, the prototype of the proposed approach was used to automatically integrate the contemporaneous records with the preliminary schedule that was developed by the contractor at the start of the project. To model schedule changes, the updated schedule that was issued at a later date to accommodate the alignment check on the TBM was used. As the simulation progressed

and once a schedule update became available, the schedule network, activity durations and schedule constraints were updated, while preserving analysis results, as per the schedule update.

The activity information window provides a tabular view of the project activities on the left and the schedule information associated with a selected activity on the right (Figure 7). The predecessor and successor tabs on the bottom right show the activity relationships driving the calculation, while the impacting events tab shows the events that have impacted the activity, if any. As can be noted from Figure 7, tunneling through the first 68 m of sandstone was impacted by three events, which resulted in a five-day increase in the planned duration.

FORENSIC SCHEDULE INFORMATION MODELING (FORSIM)										
Setup Activities Info Analysis Impacting Events Critical Path Info Impact Quanti	fication Tabulated View									
PROJECT ACTIVITIES PLANNED SIMULATE										
Sandstone 68 mSandstone curve 32 mTBM Survey CheckSandstone curve 15 mSandstone 62.09 mMobilize to sitePrepare shaft linerExcavate shaft (4.8 m)Continuously drill the shaft (20 m)Build safety wall and ladderComplete shaft constructionS-4-2018 8:00 AMS-4-2018 8:00 AMLaser move & TBM MechanicS-8-2018 8:00 AMLaser move & Crane adjustmeS-9-2018 8:00 AMLaser moveLaser moveLaser moveLaser moveLaser moveLaser moveLaser move<	n: 18 r: [ProjectCalendar 2018-04-23 8:00:00 AM 2018-05-16 8:00:00 AM 2018-05-16 8:00:00 AM 2018-05-16 8:00:00 AM 0 Event Duration al Issue 1 ent 1 3									

Figure 7. Simulated activities information for the tunneling segment.

5.3. Delay Analysis Facilitation

One of the main objectives when analyzing delay claims is establishing a factual matrix and a precise chronology of events that impacted project activities and, subsequently, overall project schedule. A review of the as-built schedule provides an overall view of the delays to the project from the planned start date to the finish date. To analyze the activities of the as-built schedule, it is important to identify the critical activities that drive the schedule. However, critical activities on the last portion of the date on which the schedule status is being reported (data date) cannot be identified through the use of conventional scheduling tools. Consequently, the driving activities of an as-built schedule or schedule updates. In the absence of schedule updates, the identification of controlling activities becomes a subjective process that is based on the opinion of project participants. In such cases, the proposed approach uses the simulation traceability feature to track and demonstrate controlling activities of completed work based on a predefined total float specified by the user or analyst, or through the calculation of the lowest total float during the simulation of the entity instance.

Figure 8 illustrates bar charts for controlling and critical activities at the time of analysis. For each activity, the planned duration and simulated duration are plotted with blue and red bars, respectively. Details of the planned duration and simulated duration of an activity



can be observed by hovering a cursor over the desired activity bars. This helps analysts identify and isolate controlling activities with large delays and allows for more focused delay analysis.

Figure 8. Illustration of the controlling activities of the tunneling segment.

One form of analysis is based on an as-planned versus as-built schedule delay analysis method. It is a retrospective approach used to compare the baseline, or as-planned, construction schedule against the as-built schedule. The as-built schedule reflects the progress of all activities and milestones throughout the project and provides verification of the driving activities of the schedule. The proposed approach facilitates as-built vs. as-planned analysis by automatically providing activity information (planned and actual dates), thereby identifying controlling and critical activities, along with a detailed documentation of events that impacted activity progress, in one way or another.

Some of the other observed benefits of the proposed approach to as-built schedule development and analysis include:

- The method takes into account the uncertainty of event timelines. Previous studies show that the data capturing process is subject to human errors, the unavailability of contemporaneous records, and conflicting information regarding the progress of an activity due to inaccuracies and inconsistencies between the contemporaneous records available for a project. More specifically, in this case study, some daily site reports were not available because they were never sent, or, in some instances, the wrong reports were sent (for instance, the report of the previous date was sent a second time, resulting in duplicate reports). This negligence resulted in uncertainty with the timelines used when developing the event schedule. For these cases, the proposed framework enables the use of distributions to model timelines uncertainties; for this case study, uniform distributions were used.
- The method limits the level of scheduling skills required for the development of as-built schedules. Developing realistic as-built schedules requires experienced schedulers to review project documentation and establish the activity execution sequence. The current system eliminates this subjective process and, in contrast, helps inexperienced schedulers have a better understanding of projects.
- The proposed method limits the need for balancing between purpose of analysis and as-built schedule preparation cost. Typically, delay analysts are constrained by the need to

strike a balance between the objective of the analysis and the cost of preparing an as-built schedule. For instance, in the case of retrospective delay analysis, the time required to identify the start and finish dates for all activities on a large schedule is disproportionate in terms of the cost and time required, especially when delay events impacted a small portion of the activities. Practically, therefore, analysts compromise by collapsing activities that are pertinent to the analysis process into a single activity or bar when illustrating as-built schedules. When such a balance is made, analysts are required to demonstrate that the omission of identifying actual dates of non-critical work was not intentional to avoid contradictory or negative evidence that does not support the conclusion of the analysis. On the contrary, the proposed framework eliminates the need of this subjective and judgmental balancing process by enabling analysts to focus on identifying the actual dates of events that might have impacted the project schedule, rather than wasting effort to identify actual dates for all activities.

• The proposed method offers a more objective approach to as-built development: Traditionally, as-built schedules are created either from scratch or a fully progressed schedule update and are then modified or augmented as needed. Both approaches require analyst interaction with the schedule, which makes the development process subjective and easy to manipulate. The proposed framework eliminates any interaction with the schedule updates, when available, with contemporaneous records, while simultaneously modeling the dynamics of schedule changes.

5.4. Results Validation

To validate results, a comparison between the simulated planned dates with those of the baseline schedule and schedule update were drawn. The tabulated view window, as shown in Figure 9, provides a tabular view of tunneling segment activities, reflecting both the simulated project plans and as-built records. The timelines of the "Sandstone 68 m" activity, as well as the planned early start date of "Sandstone curve 32 m" activity, are matched with those indicated in the baseline schedule (Figure 5) primarily because the simulation model was initially based on the baseline schedule. Once the schedule update became available, the network model was updated to reflect schedule changes (TBM alignment check). Consequently, the timelines of the remaining activities are matched with those indicated in the updated schedule (Figure 6).

FORENSIC SCHEDULE INFORMATION MODELING (FORSIM)											
Setup Activities Info	Analysis Im	pacting Events	Critical Path Info	Impact Quantific	ation Tabulated	d View					
Activity Plan	ned Duration	Planned ES	Planned EF	Simulated Duration	Simulated ES	Simulated EF					
Sandstone 68 m	13	4-23-2018 8:00 AM	5-9-2018 5:00 PM	18	2018-04-23 8:00:00	2018-05-16 8:00:00					
Sandstone curve 32 m	12	5-10-2018 8:00 AM	5-25-2018 5:00 PM	13	2018-05-17 8:00:00	2018-06-04 8:00:00					
TBM Survey Check	25	5-28-2018 8:00 AM	6-29-2018 5:00 PM	25	2018-06-05 8:00:00	2018-07-09 8:00:00					
Sandstone curve 15 m	6	7-2-2018 8:00 AM	7-9-2018 5:00 PM	8	2018-07-10 8:00:00	2018-07-19 8:00:00					
Sandstone 62.09 m	12	7-10-2018 8:00 AM	7-25-2018 5:00 PM	15	2018-07-20 8:00:00	2018-08-09 8:00:00					
Mobilize to site	10	5-10-2018 8:00 AM	5-23-2018 5:00 PM	11	2018-05-17 8:00:00	2018-05-31 8:00:00					
Prepare shaft liner	4	5-24-2018 8:00 AM	5-29-2018 5:00 PM	4	2018-06-01 8:00:00	2018-06-06 8:00:00					
Excavate shaft (4.8 m)	4	5-30-2018 8:00 AM	6-4-2018 5:00 PM	4	2018-06-07 8:00:00	2018-06-12 8:00:00					
Continuously drill the shaft (20	m) 8	6-5-2018 8:00 AM	6-14-2018 5:00 PM	8	2018-06-13 8:00:00	2018-06-22 8:00:00					
Build safety wall and ladder	4	6-15-2018 8:00 AM	6-20-2018 5:00 PM	4	2018-06-25 8:00:00	2018-06-28 8:00:00					
Complete shaft construction	1	7-26-2018 8:00 AM	7-26-2018 5:00 PM	1	2018-08-10 8:00:00	2018-08-10 8:00:00					
▲					<	<					

Figure 9. Tabulated view of simulated activities information.

Moreover, the simulated as-built timelines can be validated at an activity-level by comparing event information with simulated timelines. As it can be noticed in Figure 7, some of the planned activity durations changed after the simulation and integration of event(s) that impacted the activities and resulted in extension to the planned completion dates. For instance, the "Sandstone 68 m" activity, as shown in Figure 7, was impacted by three events that contributed a total of 5 days of delay to its planned completion date. It is also noticeable that the durations of un-impacted activities remained unchanged, while the start and finish timelines were automatically updated based on the modeled CPM calculation. In general, increase in activities durations indicate delay while decrease in activities durations indicate acceleration. These comparisons show that the proposed approach can successfully generate as-built schedules. It also provides schedule analysts with information needed to better understand project execution. It is important to note that the prototype is currently limited to one calendar per schedule due to technical difficulties with the MPJX library, which can easily be addressed in future implementation and development.

6. Conclusions

The role of as-built schedules in claims analysis cannot be underestimated. While existing methods for as-built schedule development have been gradually improving in terms of data capture efficiency, automation of schedule generation, and visualization, the existing methods limit the usefulness of as-built schedules, prolong the development process, and increase cost. Previous studies also fail, collectively, to provide desired features of as-built schedules, which would ultimately add to their value in claims analysis. This study introduced a simulation architecture to automatically develop as-built schedules that are specifically tailored to facilitate the delay analysis process. The simulation approach enables an efficient data integration process and adds valuable features to as-built schedules, such as event traceability, data organization, and uncertainty modeling. Due to the importance of these features in delay analysis, having them as mandatory requirements in claims submission is long overdue. In this method, as-built schedules can be developed either prospectively or retrospectively with limited scheduling knowledge and expertise. The proposed data structure for the schedule of events is simple, logical, and helps project teams to organize project data in a manner that is useful for many project management functions. Additionally, the modeling approach is flexible and can easily be adopted for modeling other complex construction operations. A prototype model was also developed, and real project data were used to demonstrate framework capabilities and for validation purposes.

This study showed that simulation technology and principles can be beneficial in integrating and analyzing project data including schedule updates. As the present study served as a proof-of-concept, expansion to the framework to cover the delay analysis process will be explored in future studies. This expansion includes a delay analysis mechanism, events categorization, liability allocation, and integration of model-based impact quantification.

Author Contributions: Conceptualization, M.F., Y.M. and S.A.; Methodology, M.F. and Y.M.; Software, M.F.; Validation, M.F.; Resources, S.A.; Writing—original draft, M.F.; Writing—review & editing, Y.M. and S.A.; Supervision, Y.M. and S.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Some or all data, models, or code generated or used during the study are proprietary or confidential in nature and may only be provided with restrictions (e.g., anonymized data).

Conflicts of Interest: The authors declare no conflict of interest.

References

- 1. Knoke, J.R.; Jentzen, G.H. Developing an as-built schedule from project records. *Trans. AACE Int.* **1996**, CS31–CS34.
- Hegazy, T.; Elbeltagi, E.; Zhang, K. Keeping Better Site Records Using Intelligent Bar Charts. J. Constr. Eng. Manag. 2005, 131, 513–521. [CrossRef]

- 3. Henschel, B.A.; Hildreth, J.C. Schedule Impact Analysis Using CPM Schedules; Virginia Tech: Blacksburg, VA, USA, 2007.
- 4. Avalon, A. Calculating the As-Built Critical Path. In Proceedings of the 2014 AACE International Annual Meeting, New Orleans, LA, USA, 5–18 June 2014; Long International: New Orleans, LA, USA, 2014.
- 5. Hegazy, T.; Menesi, W. Critical Path Segments Scheduling Technique. J. Constr. Eng. Manag. 2010, 136, 1078–1085. [CrossRef]
- 6. Kahler, D. Automated development of as-built construction schedules. In Proceedings of the ASCE Texas Section Fall 2012 Conference, Fort Worth, TX, USA, 7–10 November 2012.
- Mbabazi, A.; Hegazy, T.; Saccomanno, F. Modified But-For Method for Delay Analysis. J. Constr. Eng. Manag. 2005, 131, 1142–1144. [CrossRef]
- Chin, S.; Yoon, S.; Choi, C.; Cho, C. RFID+4D CAD for Progress Management of Structural Steel Works in High-Rise Buildings. J. Comput. Civ. Eng. 2008, 22, 74–89. [CrossRef]
- Rebolj, D.; Babic, N.C.; Magdic, A.; Podbreznik, P.; Pšunder, M. Automated Construction Activity Monitoring System. *Adv. Eng. Inform.* 2008, 22, 493–503. [CrossRef]
- Golparvar-Fard, M.; Peña-Mora, F.; Savarese, S. Application of D4AR—A 4-dimensional augmented reality model for automating construction progress monitoring data collection, processing and communication. J. Inf. Technol. Constr. 2009, 14, 129–153.
- Ibrahim, Y.M.; Lukins, T.C.; Zhang, X.; Trucco, E.; Kaka, A.P. Towards Automated Progress Assessment of Workpackage Components in Construction Projects Using Computer Vision. *Adv. Eng. Inform.* 2009, 23, 93–103. [CrossRef]
- 12. Zhang, X.; Bakis, N.; Lukins, T.C.; Ibrahim, Y.M.; Wua, S.; Kagioglou, M.; Aouad, G.; Kaka, A.P.; Trucco, E. Automating Progress Measurement of Construction. *Autom. Constr.* 2009, *18*, 294–301. [CrossRef]
- 13. Turkan, Y.; Bosche, F.; Haas, C.T.; Haas, R. Automated progress tracking using 4D schedule and 3D sensing technologies. *Autom. Constr.* **2012**, 22, 414–421. [CrossRef]
- 14. Hegazy, T.; Abdel-Monem, M. Email-based system for documenting construction as-built details. *Autom. Constr.* **2012**, 24, 130–137. [CrossRef]
- 15. Hegazy, T.; Ayed, A. Neural Network Model for Parametric Cost Estimation of Highway Projects. *J. Constr. Eng. Manag.* **1998**, 124, 210–218. [CrossRef]
- Memon, Z.A.; Abd-Majid, M.Z.; Yusoff, N.I.M.; Mustaffar, M. Systematic Approach for Developing As-built Schedule for Construction Project. J. Kejuruter. 2006, 18, 135–146.
- 17. Elazouni, A.; Salem, O.A. Progress monitoring of construction projects using pattern recognition techniques. *J. Constr. Manag. Econ.* **2010**, *29*, 355–370. [CrossRef]
- Alkass, S.; Mazerolle, M.; Tribaldos, E.; Harris, F. Computer-Aided Construction Delay Analysis and Claims Preparation. *Constr. Manag. Econ.* 1995, 13, 335–352. [CrossRef]
- Zhu, H.; Hwang, B.-G.; Ngo, J.; Tan, J.P.S. Applications of Smart Technologies in Construction Project Management. J. Constr. Eng. Manag. 2022, 148, 04022010. [CrossRef]
- Fagiar, M.; Mohamed, Y.; AbouRizk, S. Simulation-based Framework for Construction Delay Analysis. In Proceedings of the 2019 CSCE Annual Conference, Laval, QC, Canada, 12–15 June 2019.
- 21. Long, R.J. As-Built but-for Schedule Delay Analysis; Long International, Inc.: New Orleans, LA, USA, 2018.
- 22. AACE International. Forensic Schedule Analysis-TCM Framework: 6.4—Forensic Performance Assessment; AACE International, Inc.: Morgantown, WV, USA, 2011.
- 23. Navon, R.; Haskaya, I. Is detailed progress monitoring possible without designated manual data collection? *Constr. Manag. Econ.* **2006**, *24*, 1225–1229. [CrossRef]
- 24. Hegazy, T.; Abdel-Monem, M.; Saad, D.A. Framework for Enhanced Progress Tracking and Control of Linear Projects. *Eng. Constr. Archit. Manag.* 2014, 21, 94–110. [CrossRef]
- Abdel-Monem, M.; Hegazy, T. Enhancing Construction As-Built Documentation Using Interactive Voice Response. J. Constr. Eng. Manag. 2013, 139, 895–898. [CrossRef]
- Golparvar-Fard, M.; Peña-Mora, F.; Savarese, S. Automated Progress Monitoring Using Unordered Daily Construction Photographs and IFC-Based Building Information Models. J. Comput. Civ. Eng. 2015, 29, 04014025. [CrossRef]
- Park, J.; Kim, J.; Lee, D.; Jeong, K. Deep Learning–Based Automation of Scan-to-BIM with Modeling Objects from Occluded Point Clouds. J. Manag. Eng. 2022, 38, 04022025. [CrossRef]
- 28. Jeong, H.D.; Gransberg, D.D.; Shrestha, K.J. Framework for Advanced Daily Work Report System; Institute for Transportation, Iowa State University: Ames, IA, USA, 2015.
- 29. Shrestha, K.J.; Jeong, H.D. Computational algorithm to automate as-built schedule development using digital daily work reports. *Autom. Constr.* 2017, 84, 315–322. [CrossRef]
- Al-Gahtani, K.; Al-Sulaihi, A.A.; Iqupal, A. Total Float Management: Computerized Technique for Construction Delay Analysis. *Can. J. Civ. Eng.* 2016, 43, 391–401. [CrossRef]
- Son, H.; Kim, C.; Cho, Y.K. Automated Schedule Updates Using As-Built Data and a 4D Building information Model. J. Manag. Eng. 2017, 33, 04017012. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.