DELAYMASTER: AN INTRODUCTION

1. The Problem

Legal disputes are a major financial drain on construction projects worldwide. Delay disputes account for a large portion of overall project dispute resolution costs, which in turn are a significant contributor to overall construction costs. Construction accounts for around 15% of world GDP.

Simply put, a 'delay dispute' is a disagreement between parties to a construction contract (typically the contractor and the owner/ developer) as to which side is or was at fault for causing delay to the completion of a project (or a part of a project).

Major construction contracts usually require contractors to deliver a project by an agreed completion time. This obligation is secured by liquidated damages: a pre-agreed monetary sum to be deducted from payments due to the contractor for each day that the project overruns its schedule. Under typical arrangements, the contractor is relieved from damages for delays that are not its fault and for other types of delays that the contract designates as 'excusable'. Relief is provided by extensions of time (EOTs) that postpone the time by which the contractor is obliged to complete the works. EOTs are decided and awarded by an umpire-like third party (known as the 'superintendent' in Australia).

To establish EOT entitlement it is necessary for a contractor to demonstrate the following:

- Whether on a proper interpretation of the contract, the event or process that caused the delay is 'excusable';
- Whether the processes or events that caused the excusable delay occurred on the critical path (and thus caused delay to the completion time of the whole project); and
- The amount of excusable critical delay.

Other factors complicating this inquiry include as follows:

- If a delay is excusable, the question of whether or not it is also 'compensable' and thereby gives rise to an entitlement to prolongation costs.
- Whether or not concurrent delay has occurred, and if so, how EOT and cost entitlements should be resolved in the circumstance.
- Whether, on a proper interpretation of the contract, the contractor 'owns the float'. This influences the formula used to derive EOT entitlement.
- Whether the project has staged or sectional completion milestones (which multiplies the number of separate analyses required).

2. **Methods of Delay Analysis**

In the construction industry, project time schedules are usually modelled by a 'critical path network' via the 'critical path method' (CPM). CPM was first introduced in the late 1950s and became well-known in the 1960s. With the introduction of CPM software for personal computers in the 1980s, CPM became entrenched within the construction industry.

Methods of delay analysis have evolved in parallel with CPM; they were first documented in the 1960s, and became more widely used by the 1980s. By the 1990s, methods of delay analysis were documented in textbooks. In the 2000s, methods used within the industry were formally described in guidelines published by the *Society of Construction Law (UK)* (SCL) and the US-based *Association for the Advancement of Cost Engineering International* (AACEI). The two resultant documents are known as the 'SCL Protocol' and 'AACEI Recommended Practice 29R-03') respectively.

The SCL Protocol outlines six alternative methods of delay analysis. The AACEI RP details outlines nine alternatives. These methods are crude, pseudoscientific, obscurantist, opaque and lacking in mathematical rigour. They trace their genesis to the 1960s when analysis was often conducted with pen and paper. Remarkably, there has been very little innovation in this area for several decades.

The SCL and AACEI methods generally contemplate that analysts create a series of static snapshots of project schedules using project management software such as *Oracle Primavera* and *Microsoft Project*. They then provide for a series of inferences to be drawn about the causality of delay from that sequence of prepared schedules.

> ***There is (until now) no specialist software for forensic delay analysis available to practitioners.***

Forensic delay analysis with these legacy methods is laborious and time-consuming. It is not very accurate. Because different methods provide models that are crude in different ways, the use of alternative methods by parties often exacerbates disputes. They are good, however, for generating consulting fee income. When matters proceed to formal dispute resolution, each party will hire experts who will prepare an analysis using their own choice of method. At trial, a judge or arbitrator will ultimately decide which expert and which technique is more credible in the case and render an opinion accordingly. On larger matters, it can take a team of professionals several months to produce a single analysis and the corresponding expert report.

The crudeness and labour intensiveness of the current methods means that practitioners cannot iterate incremental improvements to their model, nor perform sensitivity analyses on the sensitivity of input variables to outputs. This inhibits the ability of parties to narrow the issues in dispute.

Considering the sophistication of the advanced mathematical modelling tools that have been developed for other economically important sectors of industry it is astonishing that the systems in use for the resolution of the causation of delay on construction projects are as primitive and old-fashioned as they are.

> ***Forensic delay analysis as it is practised across the globe is thus ripe for disruption!***

3. Introducing *Delaymaster*

Programmed in JavaFX, *Delaymaster* is a software solution that has been designed and developed to address the problem described above. Functionally, it consists of five components:

- A graphical user interface (GUI) that allows user data to be entered, the simulation to be run and results to be communicated;
- A scheduling engine, which performs forward and backward passes across a critical path network in like manner to existing scheduling software
- A simulation engine, allowing a construction project to be simulated across multiple time spans
- Charting modules, providing graphical depictions of the results
- Reporting modules, providing transparent

Traditional project management software packages such as *Oracle Primavera* and *Microsoft Project* store scheduling data in flat databases. That makes computations and comparisons across a timeline difficult. *Delaymaster*, overcomes this problem by **object-orientation**. An object-oriented model allows data to be stored in structures that better enable multi-dimensional modelling.

Delaymaster has been developed in Melbourne, Australia but is designed to solve an industrial problem with worldwide application.

4. Mathematical First Principles

A critical path network is an adaptation of a graphical object known to mathematicians as a 'ST-network'. The 'arcs' and 'nodes' of the ST-network translate into 'milestones', 'tasks' and 'links' on a critical path network to represent the scope of work and constraints on when the work can be performed.

The critical path is the longest weighted path between the start and finish points of the schedule and is determinative of project duration. The critical path allows differentiation between delays which affect project completion, and those that do not. It is derived from a comparison of the results of two algorithms: a ‘forward pass’ to calculate ‘early times’ and a ‘backward pass’ to calculate ‘late times’.

Delay can occur both instantaneously and gradually. When a superintendent issues an instruction for additional work, for instance, the scope of work expands *instantly* and the effect on the contractor’s expected timing of the works is immediate. On the other hand, if a contractor proceeds with the works at a lower or different progress rate than anticipated or performs in a different sequence to its schedule, or if the project is held up by waiting for a task to start or a milestone to be fulfilled, the effect on timing is not instantaneous; it is *gradual*.

The insight that delay has both instantaneous and gradual forms parallels modelling techniques in other fields. In economics, for instance, fluctuations in asset values may be represented by a ‘jump-diffusion’ model. Instantaneous changes known as ’jumps’ merge with gradual ‘diffusions’ known as ’drift’.

For instance:

$$\Delta v = \int_{t=0}^{z} \frac{\partial v}{\partial t} + \sum_{n=0}^{p} k_n \delta(n) + \eta(t) dt$$

Where:

v is asset value;

$\frac{\partial v}{\partial t}$ is the (gradual) rate of change of value with respect to time;

$\delta(n)$ is a ‘Dirac delta’ or ‘unit impulse’ function modelling ‘jumps’ or instantaneous changes in value;

k_n is a constant scaling the jumps for values of n from 0 to p; and

$\eta(t)$ is a random stochastic constant representing random ‘noise’;

This equation embodies something of a universal truth: the quantitative change in a variable over time decomposes to: (1) a gradually changing component (known as ‘drift’ or ‘diffusion’), (2) an instantaneously changing component (known as ‘jumps’) and (3) a random or ‘noisy’ component.

To model the delay as it affects the completion time of a project, we may discard the noise $\eta(t)$ and change v to a second time axis, m, which forecasts future time and models the the overall project duration (or ‘makespan’). Accordingly:

$$\Delta m = \int_{t=0}^{z} \frac{\partial m}{\partial t} + \sum_{n=0}^{p} k_n \delta(n) dt$$

The gradual change component may then be replaced with a critical progress rate, r_{crit}. In practice this rate may be derived or interpolated from actual start times, actual finish times and site progress records of the activity or process on the network that is on the critical path while it is in progress and thus influences the progress rate of the whole project, so that:

$$\Delta m = \int_{t=0}^{z} r_{crit} + \sum_{n=0}^{p} k_n \delta(x) dt$$

This mathematical expression allows us to understand key flaws in the the SCL and AACEI methods.

First, it can be seen that all six SCL methods and all nine AACEI methods, model *only* the first part of this equation (i.e. the 'diffusion') or *only* the second part of this equation (i.e. the 'jumps'). In every case they provide either a jump model or a diffusion model, but not both.

Delaymaster, on the other hand, recognises that delays can be *both* instantaneous *and* gradual, and provides *both* parts of the jump-diffusion model.

Second, to minimise error, the finite interval used to model the increment the simulation time dt ought to be relatively small. As liquidated damages is generally deducted on a daily basis, hours are the appropriate unit to ensure accuracy. The SCL and AACEI diffusion methods, however, generally contemplate parsing the timescale in time windows of no less than one month. Because of the labour intensiveness of schedule preparation, smaller units of computation are not feasible. This greatly inhibits accuracy.

Delaymaster, on the other hand, automates schedule creation. In its most comprehensive setting, *Delaymaster* can accurately account for the change that occurs across *every hour* of the project.

Finally, project management software generally only calculates the critical path in respect of the overall project completion milestone. It does not do so in respect of staged or sectional completion milestones. This makes delay analysis difficult on projects that have mutliple stages or sections. *Delaymaster*, conversely, remedies this problem, and derives a separate critical path for each stage or section of the project.

5. What the Software Does

We have developed and advanced proof of concept. *Delaymaster*'s graphical user interface provides the following features:

1. A panel for specifying basic project information such and the limits of the timescales modelled

2. A spreadsheet-like interface for users to enter the parameters of the model. These include:
   - A work breakdown structure.
   - The activities that comprise the schedule and the logic links between them that represent precedence constraints.
   - The patterns of working and non-working time (known as 'calendars').
   - Instantaneous changes to be made to the network (which include adding and subtracting activities and links from the network, changing durations and lags, and changing calendars) and the events with which they are associated.
   - The finish milestones that are to be designated as termini (or staged/ sectional completion milestones)

3. A panel for *running the simulation* along with settings for 'float ownership', mode of operations and options for how concurrent delay is to be resolved.

4. An innovative set of *interactive charts* that depict the results of the simulation and its calculations graphically. These charts are animated with a slider to represent how the status of the project changes over time. Other parameters can also be changed through the GUI so that users gain visually an understanding of the sensitivity of the analysis to changes in variables. The chart views include:
   - A dynamic Gantt chart.
   - A 'float map' depicting how close to criticality each element of the network was across time.
   - A 'block and stick chart' plotting instantaneous and intervallic delays across time.
   - An 'evolution chart' depicting how the status of each activity or lagged link evolved across time.
   - A bar chart depicting the excusable, compensable and inexcusable delays that occur across time for a given completion milestone, for each event or process responsible for delay.
   - A chart of completion times relative to the passage of time.
   - A chart of concurrent delay and the path traced out by the intersection of the work fronts and the critical path (sometimes known as the "as-built critical path").

5. *Reporting modules*, to provide a level of transparency of

information suitable for forensic scrutiny. This includes:

- A description of project information and user settings
- A description of All data entered that defines the model
- Maps representing the patterns of working and non-working time for each calendar
- Simulation log
- Final results.

See the screen shots below.

Even though its primary purpose is the analysis of delay, some customers may value *Delaymaster* for its ability to communicate changes in project status graphically regardless of whether their project goes to formal dispute resolution proceedings.

6. Business Plan

We have for the most part accomplished the biggest challenge: developing the software program so as to prove the concept.

Delaymaster is one or more orders of magnitude better than alternative approaches to project delay analysis in several key dimensions: speed, accuracy, adaptability, cost and transparency.

Marketing channels would generally be the same as those for project scheduling and project management software, but with emphasis also on construction lawyers and dispute resolution practitioners worldwide.

[redacted]

This software provides an opportunity to gain a monopoly on a useful industrial tool that has worldwide scalability and virtually no distribution costs.

7. Personnel

The principal developer and owner of *Delaymaster* is Mr Anders Axelson.

Mr Axelson is one of the world's leading experts on forensic delay analysis. He has a professional background in civil engineering and law. He has extensive experience testifying as a professional expert on project delay. Mr

Axelson was a co-founder of the *Society of Construction Law Australia*. He has contributed significantly to professional literature in the field. In 2005 Mr Axelson was a co-editor of "Delay and Disruption in Construction Contracts", at the time the leading international textbook on the subject matter of forensic delay analysis.

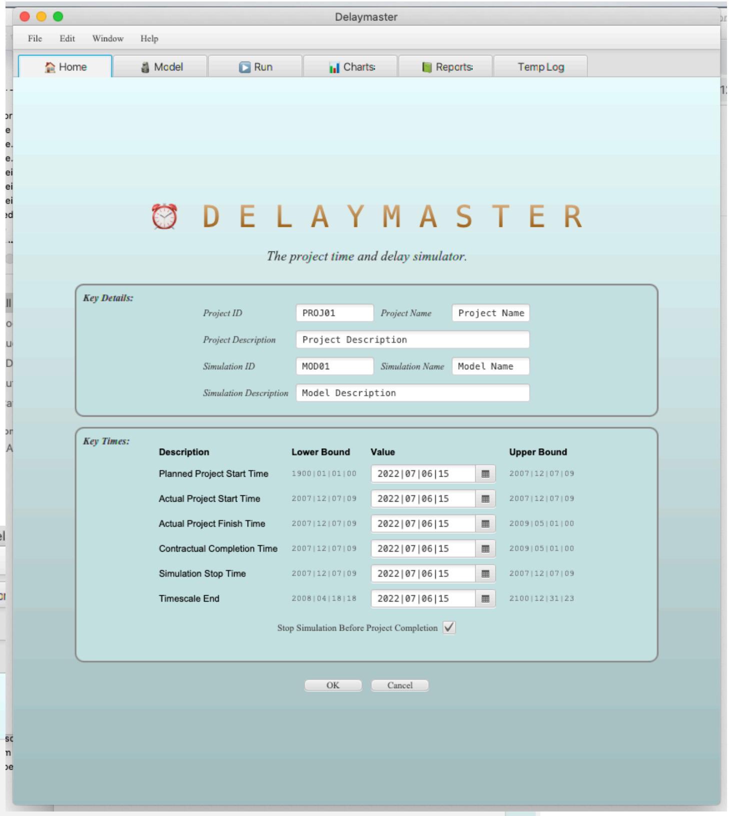
We are currently assembling a talented team to deliver on the business plan.

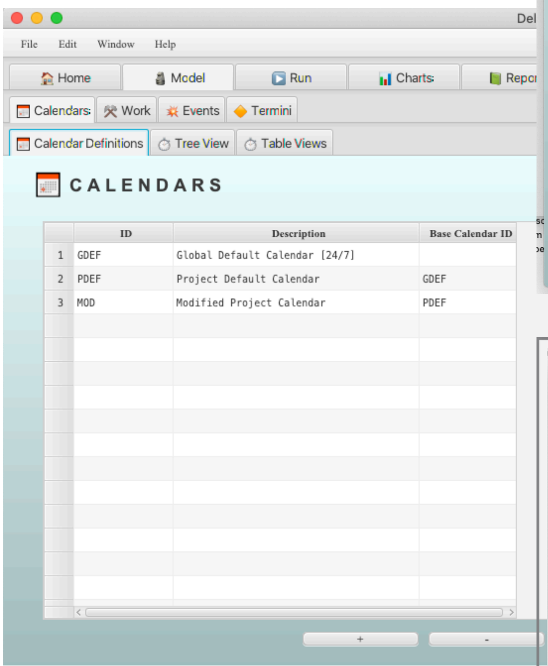
8. Further Information

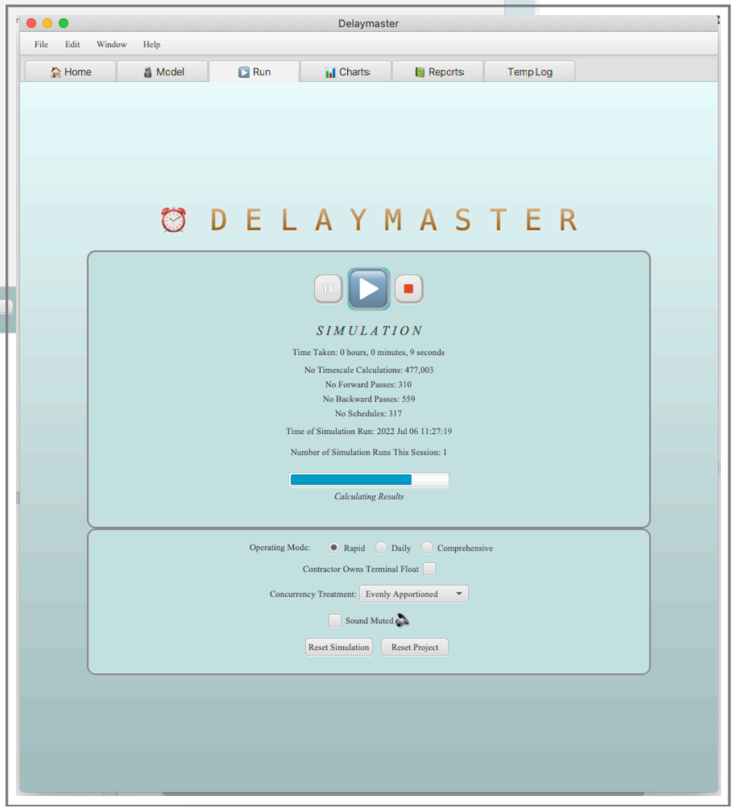
For further information please email Anders Axelson on anders@pezala.com.

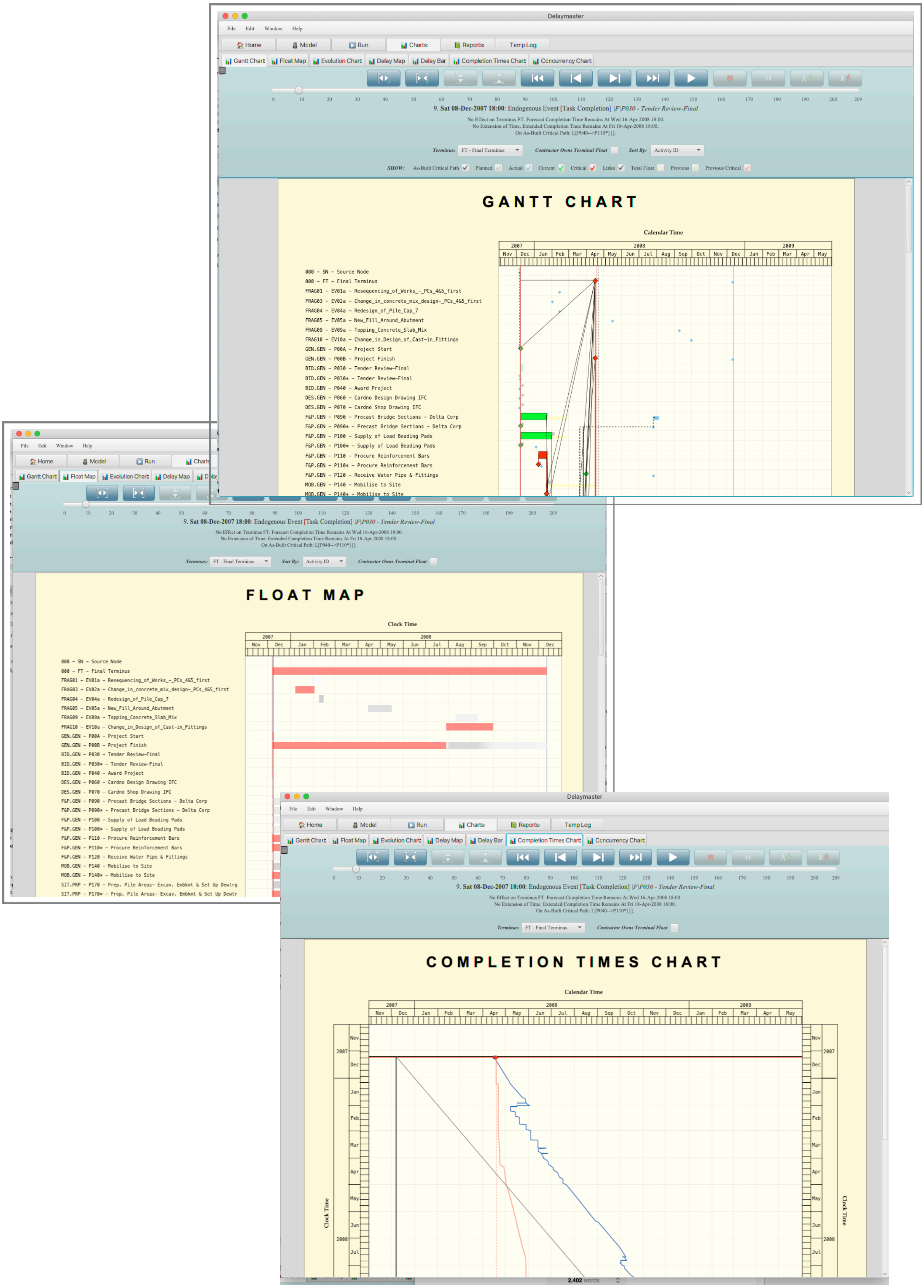
9. Screen Shots

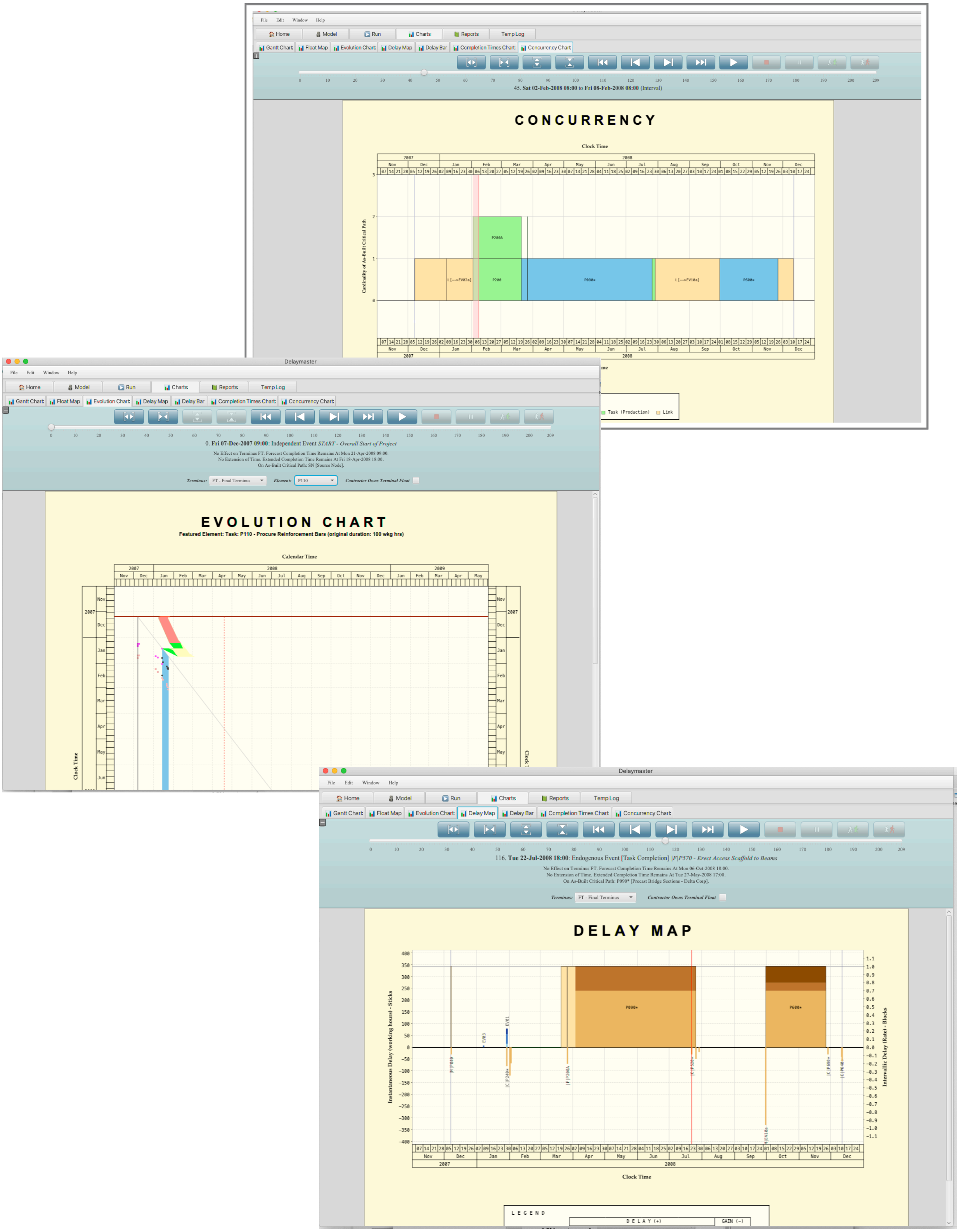
Some screen shots from the advanced proof of concept are shown below.

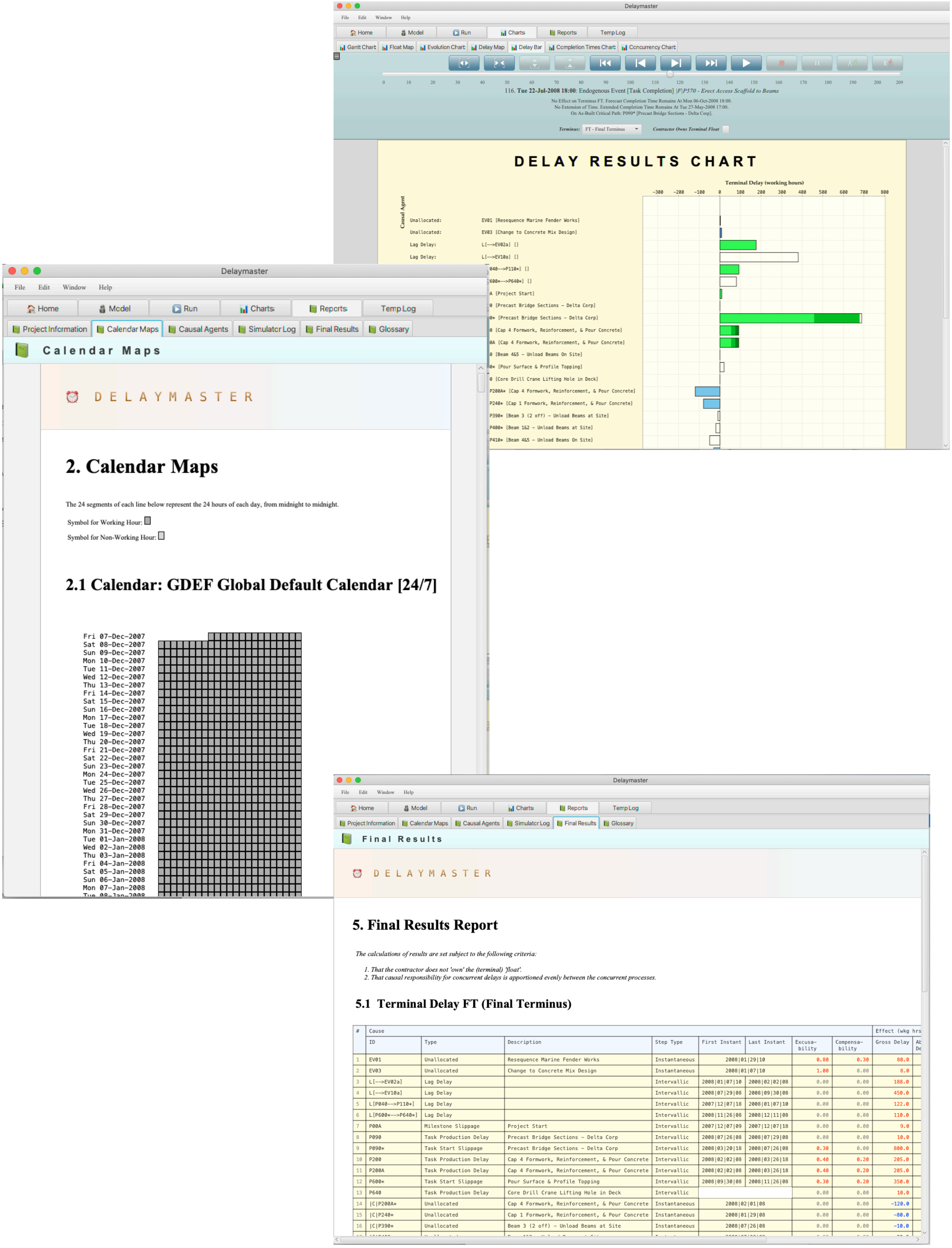

Delaymaster
File Edit Window Help
Home Model Run Charts Reports Temp Log
D E L A Y M A S T E R
The project time and delay simulator.
Key Details:
Project ID PROJ01 Project Name Project Name
Project Description Project Description
Simulation ID MOD01 Simulation Name Model Name
Simulation Description Model Description
Key Times:
Description Lower Bound Value Upper Bound
Planned Project Start Time 1900|01|01|00 2022|07|06|15 2007|12|07|09
Actual Project Start Time 2007|12|07|09 2022|07|06|15 2007|12|07|09
Actual Project Finish Time 2007|12|07|09 2022|07|06|15 2009|05|01|00
Contractual Completion Time 2007|12|07|09 2022|07|06|15 2009|05|01|00
Simulation Stop Time 2007|12|07|09 2022|07|06|15 2007|12|07|09
Timescale End 2008|04|18|18 2022|07|06|15 2100|12|31|23
Stop Simulation Before Project Completion
OK Cancel


File Edit Window Help
Home Model Run Charts Repor
Calendars Work Events Termini
Calendar Definitions Tree View Table Views
CALENDARS
ID Description Base Calendar ID
1 GDEF Global Default Calendar [24/7]
2 PDEF Project Default Calendar GDEF
3 MOD Modified Project Calendar PDEF


Delaymaster
File Edit Window Help
Home Model Run Charts Reports Temp Log
D E L A Y M A S T E R
SIMULATION
Time Taken: 0 hours, 0 minutes, 9 seconds
No Timescale Calculations: 477,003
No Forward Passes: 310
No Backward Passes: 559
No Schedules: 317
Time of Simulation Run: 2022 Jul 06 11:27:19
Number of Simulation Runs This Session: 1
Calculating Results
Operating Mode: Rapid Daily Comprehensive
Contractor Owns Terminal Float
Concurrency Treatment: Evenly Apportioned
Sound Muted
Reset Simulation Reset Project


Delaymaster
File Edit Window Help
Home Model Run Charts Reports TempLog
Gantt Chart Float Map Evolution Chart Delay Map Delay Bar Completion Times Chart Concurrency Chart
GANTT CHART
Calendar Time
FLOAT MAP
Clock Time
COMPLETION TIMES CHART
Calendar Time
Clock Time


CONCURRENCY
Clock Time
Cardinality of As-Built Critical Path
Delaymaster
File Edit Window Help
Home Model Run Charts Reports TempLog
Gantt Chart Float Map Evolution Chart Delay Map Delay Bar Completion Times Chart Concurrency Chart
EVOLUTION CHART
Featured Element: Task: P110 - Procure Reinforcement Bars (original duration: 100 wkg hrs)
Calendar Time
Clock Time
DELAY MAP
Instantaneous Delay (working hours) - Sticks
Intervallic Delay (Rate) - Blocks
Clock Time
LEGEND
DELAY (+)
GAIN (-)


Delaymaster
DELAY RESULTS CHART
Terminal Delay (working hours)
Causal Agent
Calendar Maps
DELAYMASTER
2. Calendar Maps
The 24 segments of each line below represent the 24 hours of each day, from midnight to midnight.
Symbol for Working Hour:
Symbol for Non-Working Hour:
2.1 Calendar: GDEF Global Default Calendar [24/7]
Final Results
DELAYMASTER
5. Final Results Report
The calculations of results are set subject to the following criteria:
1. That the contractor does not 'own' the (terminal) 'float'.
2. That causal responsibility for concurrent delays is apportioned evenly between the concurrent processes.
5.1 Terminal Delay FT (Final Terminus)
| Cause: ID | Type | Description | Step Type | First Instant | Last Instant | Excusa-bility | Compensa-bility | Effect (wkg hrs): Gross Delay
1 | EV01 | Unallocated | Resequence Marine Fender Works | Instantaneous | 2008|01|29|10 | 0.80 | 0.30 | 80.0
2 | EV03 | Unallocated | Change to Concrete Mix Design | Instantaneous | 2008|01|07|10 | 1.00 | 0.00 | 8.0
3 | L[-->EV02a] | Lag Delay | Intervallic | 2008|01|07|10 | 2008|02|02|08 | 0.00 | 0.00 | 188.0
4 | L[-->EV10a] | Lag Delay | Intervallic | 2008|07|29|08 | 2008|09|30|08 | 0.00 | 0.00 | 450.0
5 | L[P040-->P110*] | Lag Delay | Intervallic | 2007|12|07|18 | 2008|01|07|10 | 0.00 | 0.00 | 122.0
6 | L[P600*-->P640*] | Lag Delay | Intervallic | 2008|11|26|08 | 2008|12|11|08 | 0.00 | 0.00 | 110.0
7 | P00A | Milestone Slippage | Project Start | Intervallic | 2007|12|07|09 | 2007|12|07|18 | 0.00 | 0.00 | 9.0
8 | P090 | Task Production Delay | Precast Bridge Sections - Delta Corp | Intervallic | 2008|07|26|08 | 2008|07|29|08 | 0.00 | 0.00 | 10.0
9 | P090* | Task Start Slippage | Precast Bridge Sections - Delta Corp | Intervallic | 2008|03|20|18 | 2008|07|26|08 | 0.30 | 0.00 | 800.0
10 | P200 | Task Production Delay | Cap 4 Formwork, Reinforcement, & Pour Concrete | Intervallic | 2008|02|02|08 | 2008|03|26|18 | 0.40 | 0.20 | 205.0
11 | P200A | Task Production Delay | Cap 4 Formwork, Reinforcement, & Pour Concrete | Intervallic | 2008|02|02|08 | 2008|03|26|18 | 0.40 | 0.20 | 205.0
12 | P600* | Task Start Slippage | Pour Surface & Profile Topping | Intervallic | 2008|09|30|08 | 2008|11|26|08 | 0.30 | 0.20 | 350.0
13 | P640 | Task Production Delay | Core Drill Crane Lifting Hole in Deck | Intervallic | 0.00 | 0.00 | 10.0
14 | |C|P200A* | Unallocated | Cap 4 Formwork, Reinforcement, & Pour Concrete | Instantaneous | 2008|02|01|08 | 0.00 | 0.00 | -120.0
15 | |C|P240* | Unallocated | Cap 1 Formwork, Reinforcement, & Pour Concrete | Instantaneous | 2008|01|29|08 | 0.00 | 0.00 | -80.0
16 | |C|P390* | Unallocated | Beam 3 (2 off) - Unload Beams at Site | Instantaneous | 2008|07|26|08 | 0.00 | 0.00 | -10.0