“Enhanced” PDM Scheduling Systems

Scott C. Herold, P.E., Black & Veatch Corporation

As Presented at PMICoS Conference, April 27, 2004, © 2003 Black & Veatch Corporation

Introduction

The primary objective of a critical path method (CPM) schedule is to communicate the progress, forecasted dates, and critical paths of a project to all stakeholders. If the direct output of a project schedule is too confusing for the team to understand, or if it takes too long to prepare informative output, then the schedule fails to communicate the health of the project in a timely manner. With the advancements that have been made during the last two decades in both CPM scheduling software and the computer hardware that runs these applications, it is difficult to understand why the industry has not developed “enhanced” scheduling systems that can clearly, accurately, and quickly present the schedule data and critical paths to the stakeholders.

There are well documented factors that contribute to the difficulty in analyzing CPM schedules. These factors include the use of multiple calendars, the excessive use of constraint dates, the use of large lags in relationships, out-of-sequence progressing, inadequate and sometimes misleading output from the CPM scheduling software, and, finally, poorly conceived schedules constructed by inexperienced schedulers.

While it is easy to blame poor quality schedules on software vendors since their software allows the aforementioned “advanced” features, they are not entirely to blame. Software vendors incorporate as much flexibility as possible into their software to achieve the goal of selling the greatest number of licenses to a varied client base that employs many levels of CPM scheduling expertise.

Conversely, one could blame the inexperienced, inadequately trained, and sometimes unqualified personnel developing and managing today’s project schedules. Although there may be some merit to this claim, it does not solve the root problem: analyzing large, computerized CPM schedules can be difficult even for experienced schedulers who are knowledgeable and proficient with the scheduling software.

One solution is for the industry and the software vendors to jointly develop “Enhanced” Precedence Diagramming Method (**EPDM**) scheduling software that can not only perform all of the complex calculations that currently exist, but also has the ability to sort activities into logical “as-scheduled” paths. If software can schedule activities using multiple calendars, constraint dates, and out-of-sequence progress, then it should be capable of sorting activities into their logically scheduled sequence, regardless of the expertise of the scheduler.

The objective of this paper is to introduce and explain one set of calculations that is being used to analyze and organize CPM schedules into the “as-scheduled” critical float paths. These calculations have been developed, tested, and used in stand-alone CPM schedule analysis software since 1998.

The intent of this paper is not to teach the basic concepts of CPM scheduling; it is assumed that the reader understands early and late dates and total float calculations. Also, since the analysis tool currently in use interfaces with Primavera Project Planner (P3) Version 3.1 and because it is based on daily schedules using retained logic, the examples presented in this paper will be daily schedules calculated using retained logic in P3 Version 3.1. It is important to note that the concepts presented would apply to any type of schedule (hourly, daily, weekly), calculated with either retained logic or progress override and using any type of CPM software, provided the software follows classic CPM calculations.

Why CPM Schedule Analysis Can Be Confusing

The goal of the following exercise is to review, validate, and clearly report the critical paths of a project to the project team, without spending any time manually coding activities. The sample schedule that will be used to demonstrate the problem and explain the solution consists of 13 activities that are scheduled on three calendars, using the retained logic calculation option in P3. Calendar 1 is a five day calendar, Calendar 2 is a six day calendar, and Calendar 3 is a seven day calendar. In addition to the multiple calendars, this sample network includes non-zero lags on relationships, out-of-sequence progress, constraint dates, and float paths that have the same total float values but are independent of each other. The data date for the schedule is Monday, June 23, 2003, and the total float values for the activities vary from -14 to +5. Using the scheduling software’s existing filter, group, and sort functions, two possible layouts are built as shown on Figures 1 and 2.

Act ID	Cal ID	Orig Dur	Early Start	Early Finish	Late Start	Late Finish	Total Float
A-006	3	14	26JUL03	08AUG03	12JUL03	25JUL03	-14
A-009	3	21	26JUL03	15AUG03	14JUL03	03AUG03	-12
A-013	2	24	26JUL03	22AUG03	12JUL03	08AUG03	-12
A-007	2	12	09AUG03	22AUG03	26JUL03	08AUG03	-12
A-010	3	12	16AUG03	27AUG03	04AUG03	15AUG03	-12
A-011	2	6	23AUG03	29AUG03	09AUG03	15AUG03	-12
A-002	1	15	23JUN03	11JUL03	09JUN03	27JUN03	-10
A-005	1	15	16JUN03A	25JUL03	16JUN03A	11JUL03	-10
A-008	3	7	02AUG03	08AUG03	30JUL03	05AUG03	-3
A-004	1	5	09JUL03*	15JUL03	07JUL03	11JUL03	-2
A-003	1	5	30JUN03	04JUL03	07JUL03	11JUL03	5
A-001	1	10	09JUN03A	13JUN03A	09JUN03A	13JUN03/	
A-012	3	14	01JUN03A	14JUN03A	01JUN03A	14JUN03/	

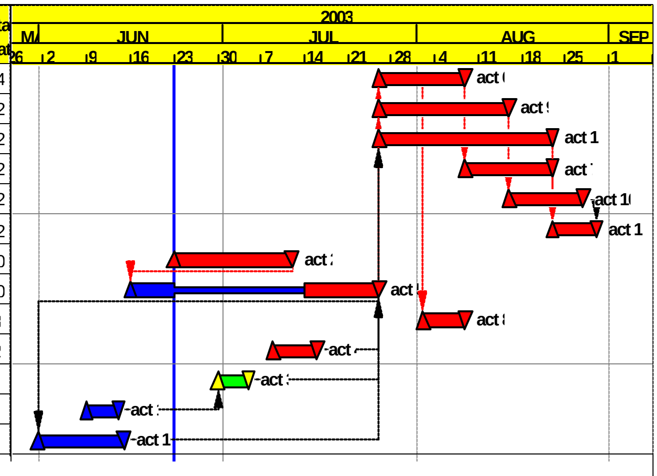

Figure 1 (Sorted by Total Float, LF, LS)

Figure 1 is sorted by Total Float, then by Late Finish, and finally by Late Start. This bar chart is difficult to analyze because it (1) mixes unrelated activities that have the same total float, (2) does not place the out-of-sequence activities in their correctly scheduled order (due to the actual dates and the blank total float values), and (3) has the tendency to move activities on the 6 and 7 day calendars closer to the top (this is because their total float values are higher than similarly delayed activities on the 5 day calendar). An activity that is 2 weeks behind schedule on a 7 day calendar has a total float of -14, whereas the same 2 week delay on a 5 day calendar is reported in the scheduling software as -10, which is shown much lower in the bar chart.

Without any manual activity coding, one can conclude that this bar chart (or a pure logic diagram or network report sorted the same way) would be difficult for a project team to analyze. The layout on Figure 1 could be improved slightly if it were grouped by calendar, and then sorted by total float and late (or early) dates. However, that layout would also be difficult to analyze if there were more than two or three calendars or if there were numerous, independent, "same-float" paths in the schedule.

Act ID	Cal ID	Orig Dur	Early Start	Early Finish	Late Start	Late Finish	Total Float
A-012	3	14	01JUN03A	14JUN03A	01JUN03A	14JUN03A	
A-002	1	15	23JUN03	11JUL03	09JUN03	27JUN03	-10
A-005	1	15	16JUN03A	25JUL03	16JUN03A	11JUL03	-10
A-006	3	14	26JUL03	08AUG03	12JUL03	25JUL03	-14
A-013	2	24	26JUL03	22AUG03	12JUL03	08AUG03	-12
A-007	2	12	09AUG03	22AUG03	26JUL03	08AUG03	-12
A-011	2	6	23AUG03	29AUG03	09AUG03	15AUG03*	-12

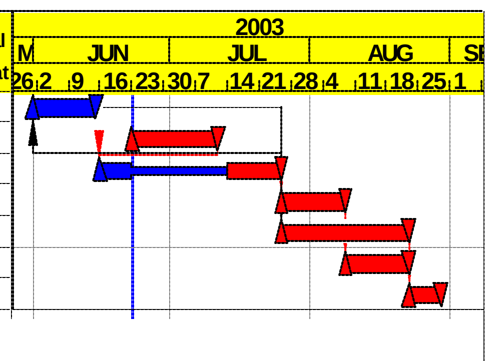

Figure 2 (Longest Path Filter, Sort by LS, LF)

Figure 2 uses the software's Longest Path filter to identify the critical path, and then it is sorted by Late Finish and Late Start. While this may seem clearer than the previous bar chart and it does show the longest path of activities, problems still exist with this layout. The main criticism is that (1) it forces the planner/scheduler to review only one float path at a time, which prevents him/her from visualizing how subcritical paths may be relating to the critical path, and (2) it forces the planner/scheduler into an inefficient analysis mode by making one change to the schedule and recalculating (often without knowing exactly how much the change has helped alleviate the schedule problem). Another potential problem with the Longest Path filter is that if the network contains activities that are scheduled later than the contract completion date of the project (e.g., the as-built drawings, warranty period work, etc.), the Longest Path filter may not retrieve the critical path of the contract completion date.

While today's scheduling software can recalculate a 5,000 activity schedule in only minutes, countless hours can be wasted using the layout shown on Figure 2 by making a single change, recalculating, refiltering, and reorganizing it over and over again. To make this inefficiency worse, more experienced and costly project controls personnel (or worse yet, whole project teams) sometimes spend hours performing this change/recalculate/refilter/reorganize sequence, since they are the only people who have enough knowledge to know what appropriate changes to make to the schedule. In the past, a clerk or technician could input changes to the schedule database, run reports for a more senior planner or project controls manager to analyze and mark up with key team personnel, and incorporate those changes in a streamlined process. Projects would be better served if experienced personnel had more time to communicate the schedules to the team, rather than personally analyzing and inputting the data.

What Does the Solution Look Like?

An EPDM scheduling system should be capable of sorting and organizing the activities into the order of criticality and the as-scheduled sort order, and should be capable of separating independent, same total float activity paths. A bar chart layout of the same activities using an EPDM scheduling system might look like Figure 3:

Act ID	Rem Dur	Cal ID	Early Start	Early Finish	Late Start	Late Finish	Total Float	RANK
0001								
A-002	15	1	23JUN03	11JUL03	09JUN03	27JUN03	-10	0001
A-005	10	1	16JUN03A	25JUL03	16JUN03A	11JUL03	-10	0002
A-006	14	3	26JUL03	08AUG03	12JUL03	25JUL03	-14	0003
A-007	12	2	09AUG03	22AUG03	26JUL03	08AUG03	-12	0004
A-011	6	2	23AUG03	29AUG03	09AUG03	15AUG03*	-12	0005
0002								
A-012	0	3	01JUN03A	14JUN03A	01JUN03A	14JUN03A		0001
A-013	24	2	26JUL03	22AUG03	12JUL03	08AUG03	-12	0002
0003								
A-009	21	3	26JUL03	15AUG03	14JUL03	03AUG03	-12	0001
A-010	12	3	16AUG03	27AUG03	04AUG03	15AUG03	-12	0002
0004								
A-008	7	3	02AUG03	08AUG03	30JUL03	05AUG03*	-3	0001
0005								
A-004	5	1	09JUL03*	15JUL03	07JUL03	11JUL03	-2	0001
0006								
A-001	0	1	09JUN03A	13JUN03A	09JUN03A	13JUN03A		0001
A-003	5	1	30JUN03	04JUL03	07JUL03	11JUL03	5	0002

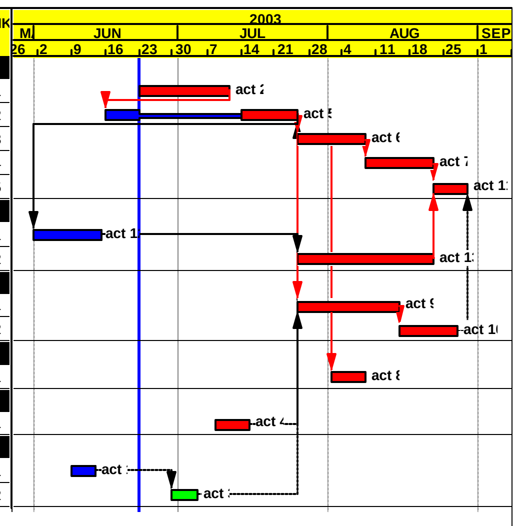

Figure 3 (Grouped by "Path" and Sorted by "Rank")

The bar chart shown on Figure 3 has many advantages compared to the bar charts shown on Figures 1 and 2. First, one can see that by looking at the "Cal ID" column (and by looking at the Total Float values), Path 0001 has activities that are scheduled on three different calendars. Although the total float values for the activities on Path 0001 are different, the EPDM scheduling system organizes the critical paths in the order that they were scheduled. Second, activities started or completed out of sequence (Activities A-005 and A-012) are placed in their proper place regardless of their actual dates or the fact that the total float column is blank for completed activities. Third, subcritical paths (ones with more positive float than the critical path) are shown separately, but are not excluded from the layout like they are on the Figure 2 Longest Path layout. Fourth, activity paths that have the same total float values, but are unrelated to each other (e.g., Paths 001, 002, and 003) are shown as separate, unrelated paths.

In addition, with this layout, one can see how a revision to the critical path may affect subpaths. For example, if the first activity on Path 0001 (Activity A-002) was statused improperly, and the correct progress was an actual start with a remaining duration of 2, it is immediately apparent that Paths 0002, 0003, and 0004 would not need to be reviewed or modified because all of those paths are successor branch subpaths to Path 0001. Paths 0002, 0003, and 0004 are automatically fixed when Path 0001 is fixed. Path 0005 (Activity A-004) would still need to be reviewed since it is NOT a successor branch subpath to Paths 0001 through 0004. The point to all of this is that more data can be clearly understood and analyzed between recalculations, refiltering, and reorganizing, and this analysis does not have to be performed by an individual sitting in front of a computer. The obvious question is "how does the EPDM scheduling system determine the path and rank codes used to organize the bar chart shown on Figure 3?"

Missing Key Data

The key concept behind EPDM scheduling software which facilitates the determination of path and rank codes, is the calculation and saving of the **relationship early and late dates** and the **relationship total float**. By treating the relationships as activities and saving the relationship dates and relationship total float, EPDM scheduling software is able to search through a CPM schedule network, identify which predecessors or successors are driving an activity's early or late dates, and clearly represent the float between two activities on different float paths.

Once the relationships are converted to activities (where the duration is equal to the relationship lag), the PDM network essentially becomes, from the computer's perspective, an Arrow Diagramming Method (ADM) network comprised of a mixture of activities and relationships, both "on-arrow." The EPDM scheduling software can now analyze this network because there are no missing links between the activities. All nodes (activities and relationships) have early dates, late dates, and total float. The EPDM scheduling system has the PDM benefits of simplicity and comprehensibility, and the ADM benefits of network traceability. Figure 4 is a diagram of the 13 activity sample network converted to an ADM network with the activities and relationships shown as arrows.

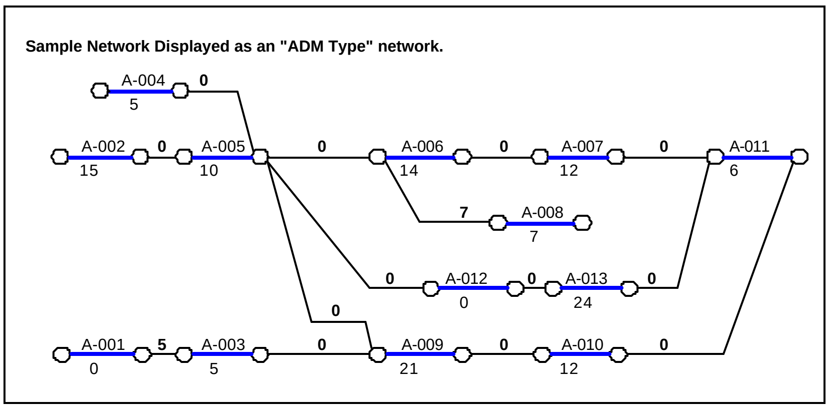

Figure 4 (PDM Network Converted to ADM Network)

To calculate relationship dates and total float, other data must be calculated first. Obviously, the additional time to calculate and save this data would slightly delay the schedule calculations, but performing these calculations would make the schedule analysis easier and faster. As a side note, if the following calculation techniques were incorporated directly into the scheduling software instead of being recalculated after activity dates are stored, the steps involved would probably be different and, undoubtedly, more efficient.

Remaining Lag

For each positive, non-zero relationship lag, the EPDM scheduling system must calculate the remaining lag that is being used for date calculations. The remaining lag is the original lag minus the difference between the data date and the predecessor's actual start or actual finish, using the predecessor's calendar. An example of a remaining lag calculation is shown on Figure 5.

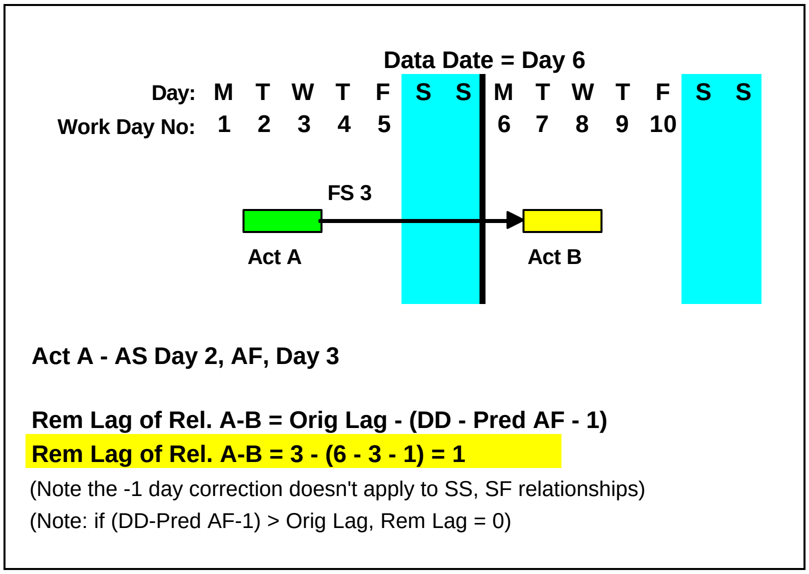

Figure 5 (Remaining Lag Calculation Example)

When CPM scheduling software calculates the early and late dates for the activities, it includes **<u>all</u>** the activities in the schedule (not just the unfinished ones), using activity remaining durations and relationship remaining lags, all starting from the project data date. Once the remaining lags are calculated, the EPDM scheduling software can calculate the relationship early and late dates and relationship total float.

Relationship Early and Late Dates and Relationship Total Float

For the next step, the early and late dates and total float for all the relationships in the schedule are calculated. To explain how relationship dates and relationship total float are calculated, the Finish-to-Start (zero day lag) relationship between Activities A012 and A013 will be used. Note that P3 Version 3.1 uses the predecessor calendar for relationship date calculations, so all of the formulas below use the predecessor calendar. The formulas are as follows:

- **Relationship ES** = Predecessor's Internal EF + 1
- **Relationship EF** = Relationship ES + Relationship Remaining Lag – 1
- **Relationship LF**= Successor's Internal LS – 1
- **Relationship LS** = Relationship LF – Relationship Remaining Lag + 1
- **Relationship TF (RTF)** = Relationship LF – Relationship EF

Note that the calculations only apply to Finish-to-Start relationships (other relationship types follow different calculations); are calculated using the predecessor's calendar; and use the predecessor/successor **<u>INTERNAL</u>** dates, not the early/late dates that P3 prints or displays. The internal early and late dates for an activity are the early and late dates of the *remaining duration* for that activity. Figure 6 shows the relationship dates calculated for the Finish-to-Start relationship between Activities A012 and A013.

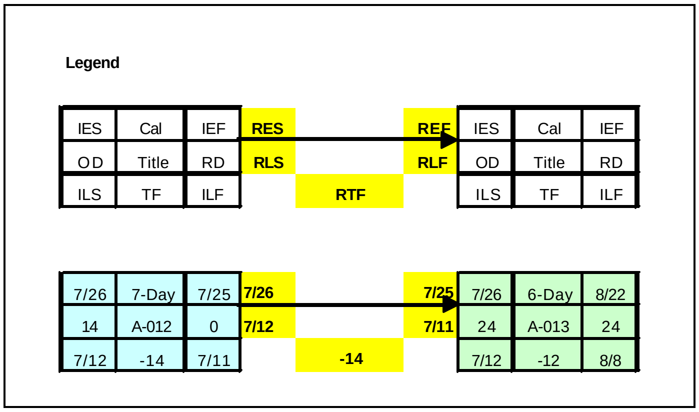

Figure 6 (Relationship Date, Relationship Total Float Example)

Resulting Successor ES/EF and Resulting Successor TF

For each relationship in the schedule, two new values must be calculated. The "resulting successor ES/EF" (RSESEF) is defined as the hypothetical early start (or early finish for FF or SF relationships) of the successor activity if that relationship was the driving relationship, using the successor calendar. The "resulting successor TF" (RSTF) is defined as the hypothetical total float of the successor activity if that specific relationship was the driving relationship, using the successor calendar. The formulas for calculating the RSESEF and RSTF for the FS 0 relationship between Activities A012 and A013 are as follows and as shown on Figure 7.

- **RSESEF** = Relationship EF + 1 (next valid workday using the successor's calendar)
- **RSTF** = Successor's Internal LS – RSESEF (using the successor's calendar)

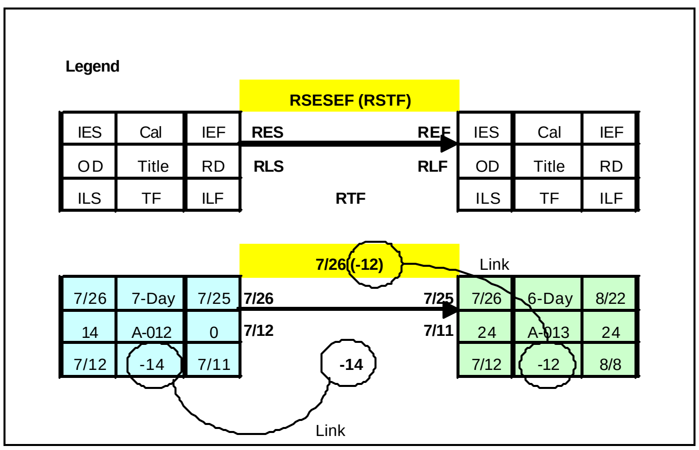

Figure 7 (RSESEF & RSTF Example)

For the reader's convenience, the relationship early and late dates, RTF, RSESEF, and RSTF for all the relationships in the entire example schedule network are shown on Figure 9 at the end of the paper.

Determination of Path and Rank

Once the relationship calculations for the entire network have been completed and saved, the EPDM scheduling system can begin identifying the most critical path and successive subcritical paths. Simplistically, the software identifies the activities that are "driving predecessors" (i.e., drive the early dates) by finding the predecessor relationships where the relationship RSTF is equal to the successor's total float. The "driving successors" (i.e., drive the late dates) are the successor activities where the relationship TF is equal to the predecessor's total float. Figure 7 shows the link between the relationship total float and the predecessor's total float, and the link between the RSTF and the successor's total float. This process continues through the entire project schedule until every activity is given a numeric path to identify its float path, and a numeric rank to identify where the activity is scheduled within the path.

With the path and rank codes for each activity identified, a custom network (predecessor/successor) report sorted by path and rank showing the activity internal dates and all the newly calculated relationship information becomes an extremely useful schedule analysis tool. The key advantages that this network report has over the standard network reports generated by P3 is that the data is already sorted by path and rank, and it shows relationship dates and relationship float on predecessors and successors.

Relationship Float versus Activity Float

When a CPM schedule has multiple calendars, constraint dates, out-of-sequence progress, and complex cross-ties between activity paths, it is important to calculate and save the relationship dates and float because the predecessor's total float values cannot be used as a quick indicator of the impact a predecessor may have on an activity; Figure 8 below is an example of this phenomenon.

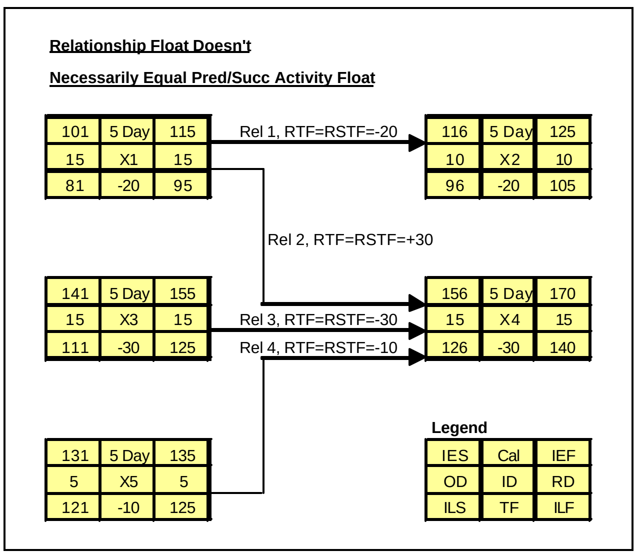

Figure 8 (Relationship Total Float may not equal Activity Total Float)

Figure 8 includes four activities scheduled on a 5 day calendar where Activities X1 and X2 are -20, Activities X3 and X4 are -30, and Activity X5 is -10. There are Finish-to-Start relationships from Activity X1 to X4 and from Activity X5 to X4. When analyzing the predecessors to Activity X4 using a P3 predecessor/successor report (or on-screen predecessor/successor window), one would see Predecessors X1, X3, and X5, and their respective total float values of -20, -30, and -10. It is obvious that Predecessor X3 is the driving predecessor (because it has the same total float value, and because the driving predecessor asterisk is present), but it is unclear what would be the "next most critical" predecessor if Predecessor X3 were deleted or progressed. In P3, the planner cannot use the total float values of Predecessors X1 or X5 to see the impact of those predecessors on Activity X4.

The only way to analyze this scenario in P3 is to examine the predecessor relationship type (Start-to-Start, Start-to-Finish, Finish-to-Finish, or Start-to-Finish), the relationship lag, predecessor/successor calendar assignments, predecessor progress (has it been finished and what is the remaining lag of the relationship), and the predecessor early start or finish (depending on the relationship type). These inputs will then be analyzed to determine which predecessor is the "next most critical." This is quite difficult to do with most CPM schedules and is precisely the reason why many people have to recalculate dates after each change; they cannot visualize what the "next most critical" path will be prior to recalculation. Using an EPDM scheduling system, one would see the relationship dates and float and would immediately know that Activity X5 would be the "next most critical" predecessor since the RSTF of -10 is less than Predecessor X1 RSTF of +30. The interesting point in this example is that the **activity total float** of Predecessor X1 is -20 whereas the **activity total float** of Predecessor X5 is only -10. Examining predecessor activity total float does not always provide a complete analysis of the problem.

Summary and Conclusions

EPDM scheduling systems that can organize themselves into "as-scheduled" float paths are within reach. Although additional calculations are required to determine path and rank codes for the activities, the value of these calculations is immeasurable. In the future, scheduling software utilizing EPDM calculations will be capable of organizing schedule data for experienced and non-experienced planners, regardless of schedule complexity. Neither multiple calendars, constraint dates, out-of-sequence progress nor large-lag relationships will prevent the software from presenting schedule information to the project team in a clear, concise manner.

The greatest benefit the industry may gain from EPDM scheduling systems is the simplification of CPM schedule analysis and delay analysis. Owners, consultants, contractors, and analysts will be able to break away from confusing schedule output and the technical aspects of the software to clearly review and discuss the critical paths of the project schedule. When this occurs, project schedule quality should improve and more attention will be focused on mitigating schedule problems and issues.

.

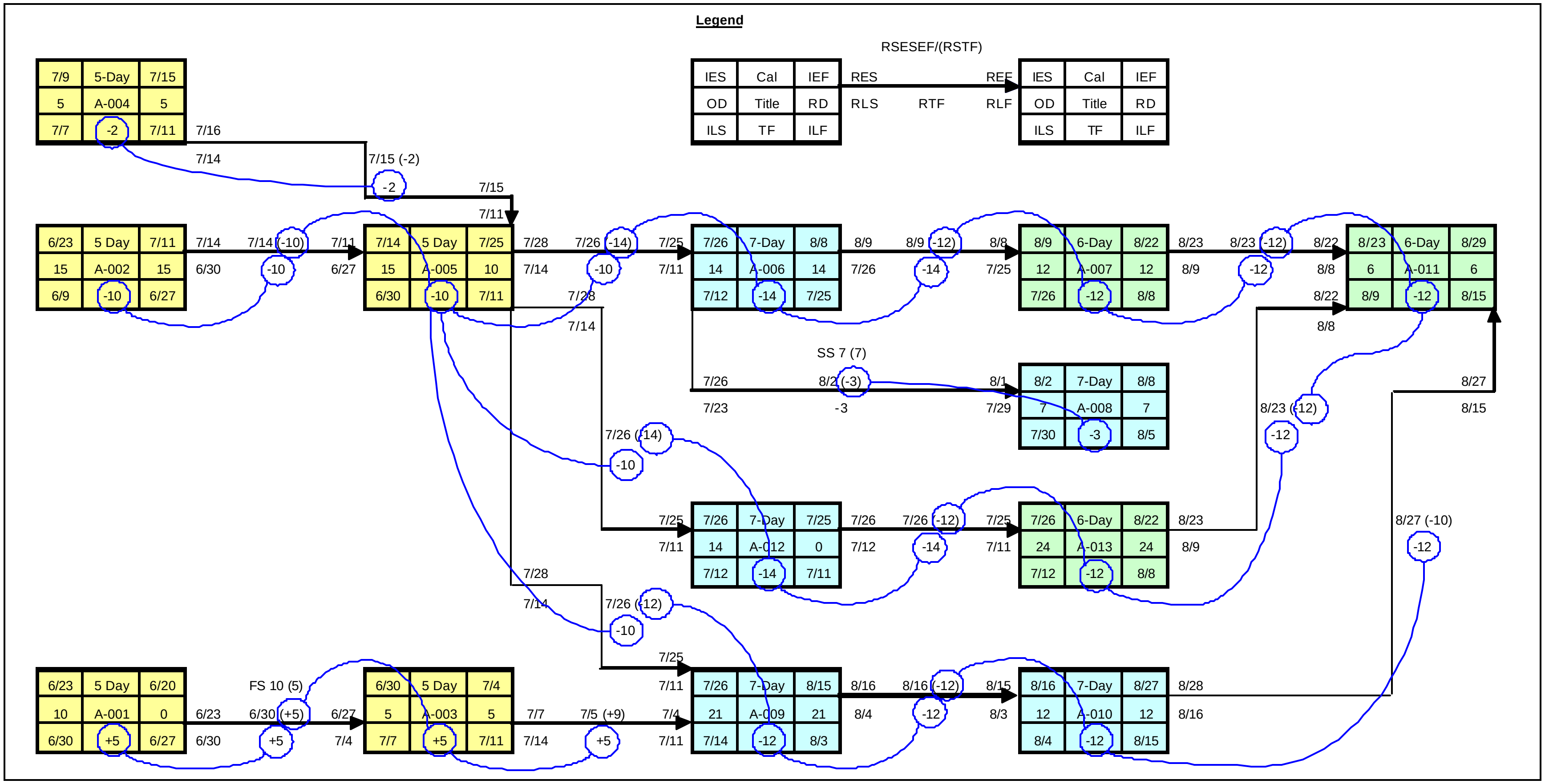

Figure 9 (Relationship Dates, RTF, RSESEF, RSTF Example)