

REVIVING A MECHANISTIC VIEW OF CPM SCHEDULES IN THE AGE OF INFORMATION TECHNOLOGY

Gunnar Lucko

Department of Civil Engineering
The Catholic University of America
620 Michigan Avenue NE
Washington, DC 20064, U.S.A.

ABSTRACT

This paper argues for reviving a more mechanistic view of CPM scheduling. It first reviews the clarity of displaying both the time and logic information of schedules in the graphics outputs of scheduling software used in construction practice. While such software offers an abundance of features, the true nature of schedules may remain hidden behind graphics defaults. Two methods introduced in the literature, fenced bar charts and a mechanical model, are then brought forth as examples of techniques supporting an intuitive structural understanding of schedules. Both stress the equal importance of activities and logic links for displaying schedules. Comparisons between these mechanistic methods and scheduling software are drawn and recommendations for achieving optimum graphical representations of schedules are provided. Eventually, a revised approach that acknowledges the complexity of schedules and the need for their unambiguous display can contribute to reducing errors and to an overall improved use of CPM scheduling.

1 INTRODUCTION

All schedules convey two basic types of information, time and logic. An activity list of activity durations and the sequencing between these activities comprises a full schedule. Graphical representations of schedules consist of four elements (Winter 1997): Activities, logic links between these activities, codes that identify and describe both the activities and their links, and additional constraints, such as e.g. resource needs and availability.

During the late 1950s the critical-path-method (CPM) was developed through efforts of DuPont and Remington Rand to create a formalized project management tool (Melin and Whiteaker 1981). CPM uses two-dimensional directional network diagrams to display activities and their logic. Two basic types of network diagrams are used in construction management, the activity-on-the-arrow

(AOA) diagram and the activity-on-the-node (AON) diagram. AOA networks are also referred to as I-J diagrams after the conventional way of naming their nodes. AON networks are also referred to as the Precedence Diagramming Method (PDM). They differ in the graphical notation, with AOA employing arrows to symbolize the dynamic nature of the activities and nodes to show their connections, while AON uses rectangular boxes or circles as the body of the activities and arrows for the logic links between them. AOA networks may require “dummy” links (displayed as dashed arrows) to capture the logic fully and correctly. A simple example of an AON schedule comprising 15 activities is listed in Table 1 and is depicted in Figure 1.

Table 1: Network Activity List

Activity	Dur.	Successors
Mob.	7	A, B, E
A	19	D, I, J
B	10	C
C	6	D, F, J
D	18	L
E	15	F, G
F	17	H, I, K
G	16	H, I, K
H	6	M
I	11	L
J	19	L
K	15	Turnover
L	18	Turnover
M	10	Turnover
T/O	3	N/A

Note: Boldface activities are on the critical path.

Despite its common use and long history, “[j]ust because CPM is over 50 years old, this does not mean that the entire width and breath [sic] of this ‘science’ has been fully defined” (Winter 2004). It comes as a surprise that “CPM scheduling does not seem to get much respect in construction. It is considered to be a burdensome, arcane tool not

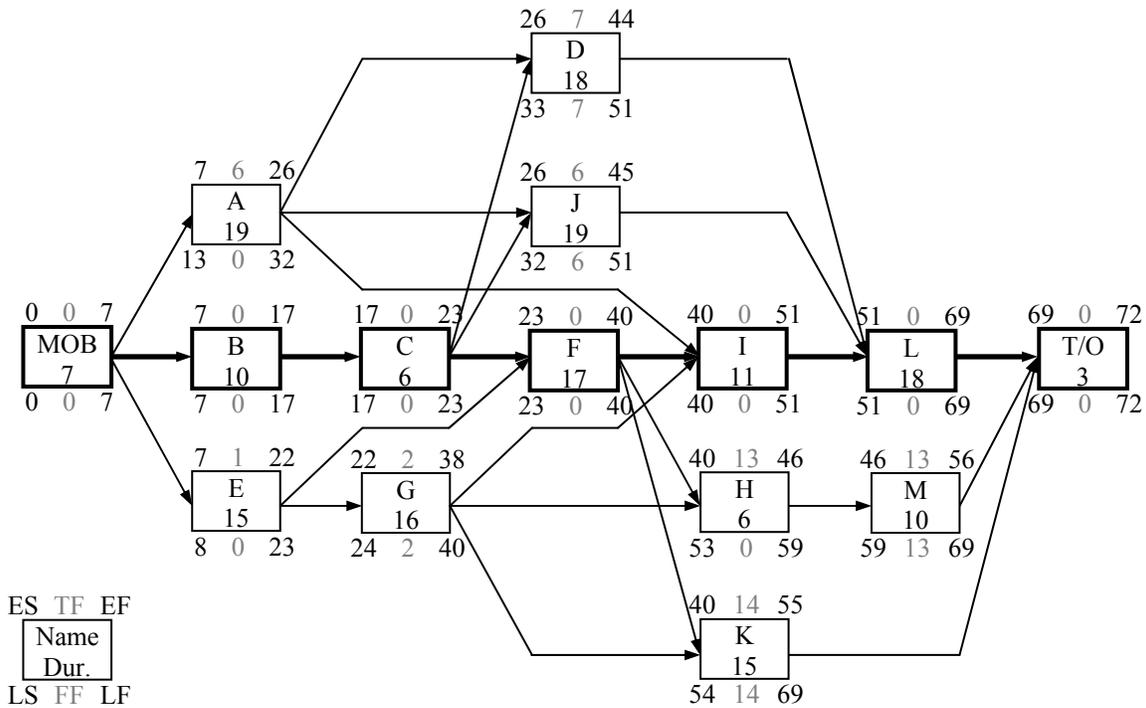


Figure 1: Sample Schedule Network with CPM Calculations

adding value to a busy project manager’s daily life” (Basu 2003). It is the goal of this paper to contribute to an increased appreciation and improved of this versatile graphical management tool.

2 CPM SCHEDULING METHODS IN SOFTWARE APPLICATIONS

Construction managers nowadays routinely rely on commercial scheduling software packages such as Microsoft® Project, Primavera Project Planner® (P3®), or Primavera SureTrak® Project Manager to develop, update, and control schedules for their construction projects and to integrate these data with other managerial functions. Software packages with their wide range of functions come with default options and wizard assistants that allow even novice users to create full construction schedules with relatively little or no training.

Each activity in the schedule is created by the user and is assigned its identifier and duration in an interactive menu. Logic information describing the relationships between activities is entered as well. Based on the overall project start the software then performs the forward pass to calculate early start (ES) and early finish (EF) and the backward pass to calculate late start (LS) and late finish (LF) dates. The software then takes their differences to calculate the total float (TF) of the paths through the network, the free float (FF) for each individual activity, and

ultimately to determine the critical path (giving the minimum overall project duration) as listed in Table 2.

Table 2: Network Activity List with CPM Calculations

Activity	Dur.	Successors	ES	LS	EF	LF	TF	FF
Mob.	7	A, B, E	0	0	7	7	0	0
A	19	D, I, J	7	13	26	32	6	0
B	10	C	7	7	17	17	0	0
C	6	D, F, J	17	17	23	23	0	0
D	18	L	26	33	44	51	7	7
E	15	F, G	7	8	22	23	1	0
F	17	H, I, K	23	23	40	40	0	0
G	16	H, I, K	22	24	38	40	2	2
H	6	M	40	53	46	59	13	0
I	11	L	40	40	51	51	0	0
J	19	L	26	32	45	51	6	6
K	15	T/O	40	54	55	69	14	14
L	18	T/O	51	51	69	69	0	0
M	10	T/O	46	59	56	69	13	13
T/O	3	N/A	69	69	72	72	0	0

Note: Boldface activities are on the critical path.

2.1 Bar Charts with Logic Links

Earliest bar charts, or Gantt charts, while well suited to display start dates, durations, and finish dates of activities on construction projects, are lacking information about the relationships between said activities. Connecting the activity bars with logic links turns simple bar charts into

full schedules that can be used to control the time aspect of construction operations. Scheduling software can routinely generate such bar charts with logic links from the activity list.

2.2 Activity-on-Node Networks

Probably the more common form of network schedules used nowadays, AON networks are fully implemented in scheduling software. In the software applications they are called PERT after the Program Evaluation and Review Technique (PERT). PERT was conceived concurrent with CPM in the late 1950s by the U.S. Navy for its Polaris missile program, whose development during the Cold War was a priority. In its original form, PERT is not another network technique, but rather integrates uncertainty into the schedule dates, whereas CPM uses static numeric dates. PERT uses three possible values for each date – one most optimistic, one most likely, and one most pessimistic. Assuming a beta probability distribution for the dates, the overall expected project duration is calculated by summing up these three values along the critical path using the formula *Expected Value = (Most Optimistic Value + 4 · Most Likely Value + Most Pessimistic Value) / 6*.

PERT in scheduling software does not use probabilistic dates but actually uses standard AON networks with static dates. The graphical representation uses activity boxes containing several rows of text, including an identifier number, name, full and remaining durations, start dates, finish dates, and resource needs, if any. Activities are arranged in a grid that may be expanded to so-called time-scaled PERT by adding a horizontal time axis. Arrows connect the activities horizontally through the network as specified by the user in the activity list menu.

3 PROBLEMS OF SOFTWARE-GENERATED SCHEDULE DIAGRAMS

Schedule networks for large projects may easily comprise thousands of individual activities, often grouped in sub-networks. Clarity in the graphical display obviously becomes increasingly important as networks become larger and more complex. Scheduling software “can produce reams of data. Interpreting this [sic] data meaningfully is a challenge for the project manager with very limited time at hand” (Basu 1990). Apart from the sheer number of activities and their logic links, complexity of schedules additionally stems from communicating information about resource constraints and possibly also financial information. In this paper, the focus is put on communication of time and logic information in scheduling software. The following sections compile various issues encountered in using the graphics output of CPM scheduling software. Suggestions for improvements are derived.

3.1 Layout and File Structure

3.1.1 Overall View

A feature included in recent releases of scheduling software is the so-called “cosmic view” of the AON network where activities are displayed as dots without any logic links. Intended to show the actual structure of the schedule, this view rather shows the arrangement of activities as generated by the software. For a structural review it would be advantageous to present both activities and logic links and carefully arrange them in the grid. Ongoing research at The Catholic University of America examines how schedules can be represented graphically to achieve optimum readability.

Scheduling software user manuals state that the time-scaled PERT schedule layout can by default hold a maximum of only three activities per column. The user can raise this number manually. Figure 2 shows how in such software-generated schedule all activities on the critical path are automatically placed into the same row of the grid with exception of the last activity. While quick identification of the critical path is thus possible it should be considered whether collapsing the rows of the schedule less than fully would improve readability of the overall schedule, as seen when comparing Figure 1 with Figure 2.

3.1.2 Electronic Storage

Unlike ubiquitous word-processing and spreadsheet software, where each document or workbook is saved in one data file that can easily be shared electronically with other users, scheduling software saves a project schedule in form of a group of numerous individual files. It could be advantageous to develop a file structure that would create only one file per schedule.

3.2 Time Information and Calendars

The default calendar axis in scheduling software is triple-scaled to years, months, and weeks. Individual weekdays or weekends are not part of the initial formatting but can be added manually by the user in a finer resolution. It remains unclear whether the tick marks or “yard lines” for weekdays indicate the duration of the particular day or its specific instance in time, i.e. whether they mark the beginning or the middle of days.

Calendar dates are used by default while ordinal dates can be enabled manually. Weekends, distinguished from weekdays by thinner activity bars or shading of calendar columns, are used in either calendar system and can be disabled manually. A functionality at the top-level menu to directly switch between these two calendar views and possibly between calendar days with weekends and pure workdays could be beneficial.

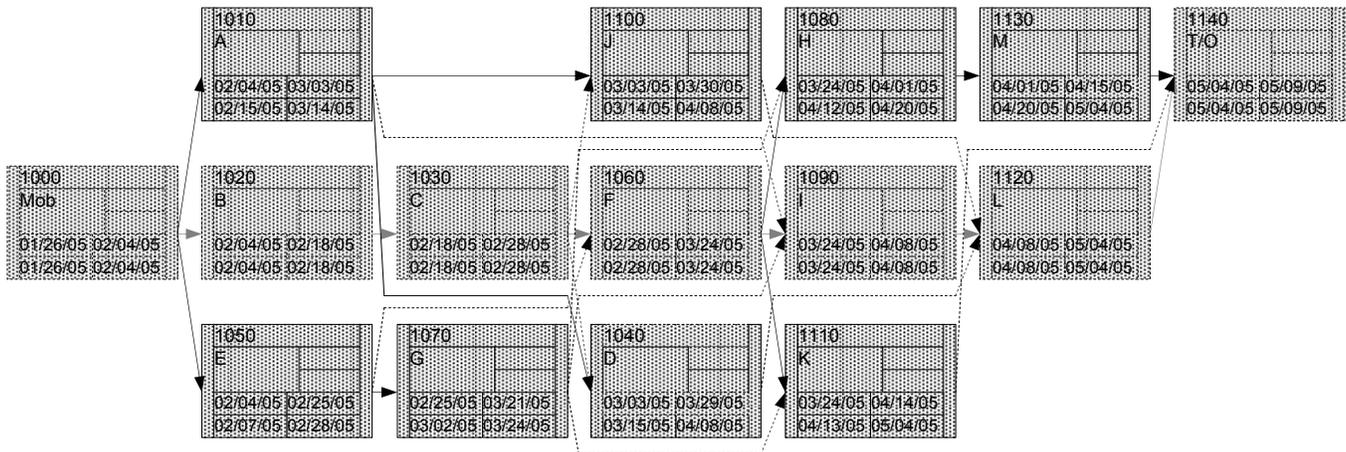


Figure 2: Software-Generated Activity-on-Arrow Diagram with CPM Calculations for Sample Schedule

Time-scaled PERT diagrams contain a time axis that is discrete, not continuous. By default each activity that starts in a particular week is grouped into the same column, which might potentially obscure sequencing between these activities, if any.

3.3 Logic Information, Activities, and Arrows

3.3.1 Activities

Both bar charts and PERT diagrams allow for a variety of customized layout options. Diamonds, baseball home plates, and flags are available to symbolize milestones and summary points. However, not all default graphics settings for bar charts with logic links follow recommendations for clear data display as put forth by Wainer (1984) and other authors.

The common way to display activities in scheduling software PERT diagrams are spacious rectangular boxes with text fields for time information. It is possible to create an activity box with rounded endpoints by specifying its percent length that shall be rounded. Arrows in PERT and time-scaled PERT are not attached directly to the leftmost and rightmost edges of the activity boxes but to the straight or rounded endpoints. Activity bars may show triangular start and finish marks called endpoints that are pointing up and down, respectively. It remains unclear whether the leftmost or rightmost edges of the triangle or its pointed tip are signifying its actual instance in time. The preformatted size of the triangle might cause a graphical deviation in the start or finish date of plus or minus one day.

A curious non-existing overlap of one day is displayed e.g. between activity B and its direct successor C in Figure 3. It is hypothesized that this is caused by an inconsistent definition of start and finish dates. Under no circumstances should start-of-day dates and end-of-day

dates be used in the same CPM calculations. This author recommends strictly using only end-of-day dates during both the forward and backward passes. An initial activity with e.g. three days duration would thus have an ES of end-of-day zero and an EF of end-of-day three. CPM calculations remain fully consistent and only simple additions and subtractions are used without any adjustments becoming necessary.

3.3.2 Arrows

Logic links and smart logic links, i.e. start-to-start, start-to-finish, finish-to-start, or finish-to-finish relationships, are displayed as arrows pointing from the predecessor to the successor activity. They can be displayed in three different ways. Direct links are use straight arrows without any bends. However, for links between the edges of two activities that are not in the same row of the grid they may potentially intersect with other activities. Shared links use one line for several links from which arrow heads branch out. Yet not even then the separate links in the scheduling software use fully individual arrows. This author recommends using a bundle of arrows that allows seeing how many successors depend on an activity by simply counting the lines at its edge and resorting to separate or shared links only if required by the number of successors or by the spacing and resolution of the overall PERT diagram.

Layout constraints of the bar chart with logic links as seen in Figure 3 appear to be the reason for partially backward pointing arrows. At first glance it may erroneously appear that activities have negative float or are connected by smart logic links whereas in reality the changing direction of arrows may simply have been caused by strictly using only horizontally forward pointing arrow heads. This author recommends using horizontally pointing arrows as far as possible but considering vertically pointing arrows if dictated by space constraints in the

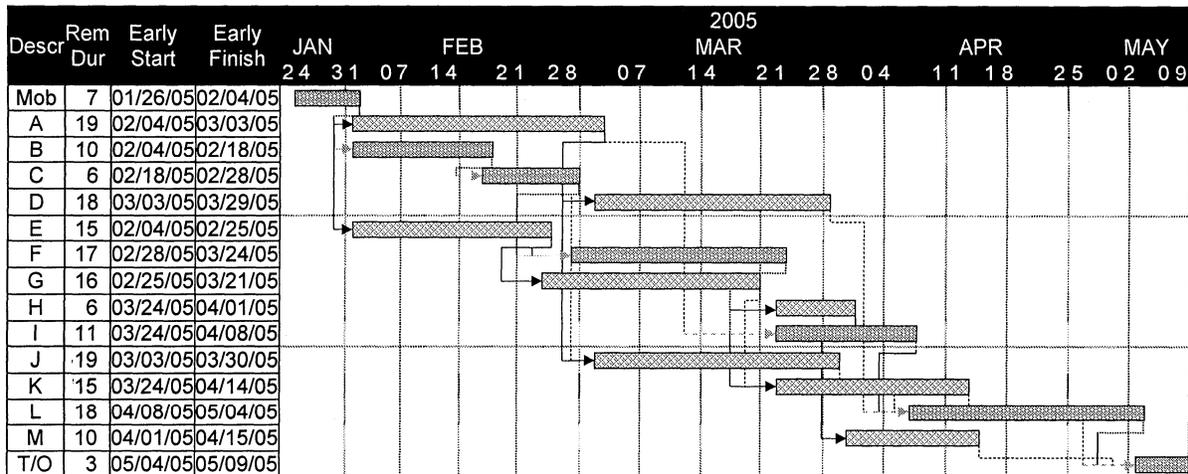


Figure 3: Software-Generated Bar Chart with Logic Links for Sample Schedule

overall diagram or when activities follow directly upon each other in a stepwise manner, e.g. on the critical path. The number of bends in the arrows should be kept at a minimum and if possible, angles and directions of arrows should follow a consistent pattern.

Based on numerous methods illustrations in scheduling software user manuals that contain only simple bar charts it appears that software developers still are focusing significantly more on activities rather than also considering logic. Fenced bar charts as introduced by Melin and Whiteaker (1981) are described in detail in the following section of this paper. Their recommendations for creating fenced bar charts have not entirely been followed by software developers as seen by comparing Figure 3, where schedule logic and float are very difficult to read, with a manually drawn fenced bar chart in Figure 4. It appears that in Figure 3 dashed lines are used to signify actual float, and solid lines are used for part of links that exist for layout reasons only. Shaded arrows and arrow heads are used to connect activities that are on the critical path.

3.3.3 Data Density

Diagrams ideally communicate the most amount of information with the least amount of ink and space used. Tufte's (2001) measure of data density, defined as the "number of entries in data matrix" divided by the "area of data graphic" can be applied to the time information of a schedule in a slightly modified version by counting the identifier label, duration, ES, EF, LS, and LF for each activity as one entry. A typical PERT diagram such as the one in Figure 2 in its original printed size is found to have a relatively low data density of about 0.5 per square centimeter for time information. Activities may be grouped into rows e.g. by activity types and columns e.g. by starting within the same week in time-scaled PERT.

Links between activities play a major role for clear display of the sequencing in the schedule. Applying the measure of data density to the logic information of the schedule, counting each arrow individually, comes out to about 5.0 per square centimeter, or a 10 times higher data density than the formatting used for activities. The interesting crosswise relationship between activities F, G, H, and K of Figure 1 for example gets lost entirely in Figure 2. Shrinking the activity size relative to the grid to provide a more generous spacing between activities where to funnel the logic links could remove this discrepancy. In general, considering the size and complexity of real construction schedules, activities and logic links should both receive equal attention when creating a schedule diagram.

3.3.4 Data-Ink Ratio

Tufte (2001) defined the data-ink ratio as the "data-ink" divided by the "total ink used to print the graphic" that can be applied to scheduling software output. The data-ink ratio should ideally approach 100%. The PERT diagram of Figure 2 could be easily improved by removing by several lines within each activity box and by removing their shading. More prominent is the superfluous ink used in Figure 3, where the title calendar bar, activity list, and texture of the activity bars could be simplified and reduced for better readability.

4 MECHANISTIC VIEWS OF CPM SCHEDULING IN LITERATURE

4.1 Fenced Bar Charts

In their classic paper Melin and Whiteaker (1981) introduced a method of graphically showing CPM schedule logic by adding vertical "fences" and horizontal "links" in a strict yet intuitive form to simple bar (or Gantt) charts.

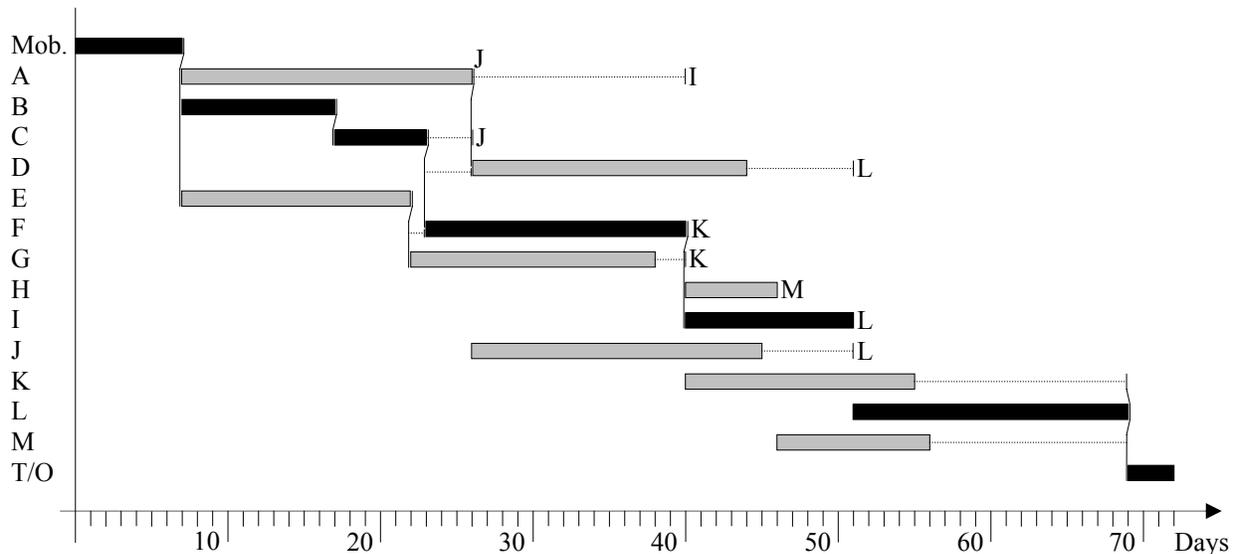


Figure 4: Fenced Bar Chart for Sample Schedule

Their method, aptly named Fenced Bar Charts, is a major contribution to clearly displaying both time and logic information, whereas previous “networks drawn range from over-simplified bar charts to walls covered with mazes of crossing and merging lines” (Melin and Whiteaker 1981). A real network is “much more complex to produce than then bar chart and typically the network tends to be non-planar with many crossovers resulting in the possibility of confusion” (Melin and Whiteaker 1981).

Fences and links are both symbolizing the mechanical nature logic links in schedules – predecessor activities “pushing” their successors further back in time in case of a delay in the operations. Summarizing and adding to the original work of Melin and Whiteaker (1981), the following conventions apply to fenced bar chart graphical notation and terminology:

- **Activity Bar:** Displaying the activity start and finish by its location and the activity duration by its length. Color-coding, dashing, shading, and texture fills offer additional degrees of freedom for visualization of information, e.g. for distinguishing between different activity groups. Tufte (2001) cautions against the use of “chartjunk” cross-hatching and Moiré fill effects that are common in computer graphics such as Figure 3 in favor of shades of gray. The critical path is often highlighted in red. This author suggests using blush shading for sub-critical activities with little TF that through minor delays would become part of the critical path.
- **Fixed Link:** Solid horizontal line displaying required time to pass between certain activities,

e.g. for curing concrete. Usually drawn after an activity bar.

- **Flexible Link:** Dashed or dotted horizontal line displaying float by its length. Usually drawn after an activity bar.
- **Fence:** Vertical solid line drawn after an activity bar and before its successors. Able to push them back in time in case of delay. Drawn to half the height of an activity bar to avoid overlapping with other fences, if any.
- **Broken Fence:** Two or more predecessors affecting an activity bar with their fences. Necessary when a logical dummy would be required in an activity-on-arrow network diagram.
- **Flag:** Using a label or pair of labels to indicate a connection in an interrupted long fence that in its original form would create crossovers. Alternatively, crossovers can be displayed by one line exhibiting a semicircular “jump” or a “break” in crossing another line.

In essence, fenced bar charts are true time-scaled network diagrams with activity durations being displayed by the length of the respective bars. Contrary to fenced bar charts, which are based on simple bar charts, time-scaled PERT layouts in scheduling software are based on activity-on-node networks. They fall short of the clarity that fenced bar charts achieve because the duration information is contained partially in the week-scaled calendar axis and partially in the numeric duration in the activity box. It is recommended to fully implement fenced bar charts in scheduling software.

Time-scaled network diagrams carry the significant advantage that concurrent activities that may compete for

available resources, e.g. labor, and are causing costs at the same time are clearly displayed in each daily column. Creating a resource-loaded or cost-loaded diagram thus simply requires adding a row summarizing the resource needs and availability or the cash flow incurred underneath the time axis. Such integrated time-money-resource diagram is an option that pure network diagrams cannot offer.

4.2 Mechanical Schedule Model

Basu (1990) created a physical schedule model that is able to simulate the forward and backward pass calculations. “Each activity is represented by a block of wood. The length of the block is proportional to the activity duration. Each end of the activity block has a pivot from which a relationship starts or finishes. A relationship is represented by a [wire] spacer bar. The board on which the activities are arranged, much like a child’s abacus, is the calendar. The CPM network is made by placing an activity on the board and connecting the activity to its successors and predecessors by the spacer bars. The calculations in the model are carried out by tilting the board” (Basu 1990) to the left for the forward pass or to the right for the backward pass. “It is important to note that the predecessor can push and not pull its successor” (Basu 1990). Adding to this definition this author notes that an exception to this rule would have to be included for smart logic links. They could physically be implemented if a small hook were added at both ends of an activity block. It is further suggested to distinguish between fixed bars for a required time passing between activities and wire springs for float.

Multiple calendars are possible in such mechanical model by breaking down each activity into “a group of one-day (time unit) blocks held together by a rubber band... The holidays are slightly ramped platforms in the calendar board such that an activity will slide aside [or apart] and span the holidays” (Basu 1990). Resource-loading the model would translate into a “stacking” (Basu 1990) summation over columns on the calendar board.

Despite the mechanical appeal of the model, the author does not provide any rules by which activities should be placed into different rows on the schedule board, but implicitly assumes that the user selects a setup that does not create much interference among spacer bars. Admittedly, a “physical model for a network of any complexity becomes difficult to build because physical intersection of relationship bars get tangled easily when a planar model is used” (Basu 1990).

This author has successfully demonstrated to students mechanical properties of simple CPM schedules by using matchboxes joined by paperclip links, both of which are readily available at any convenience store. Using matchboxes has the advantage of allowing extension of

individual activity durations by pulling out their drawer and directly seeing the delay effect on the network. This solves the problem posed by Basu (1990) for the fixed-duration wooden blocks in his mechanical CPM model. It is suggested to include into scheduling software a function that allows “tilting the board” to compare forward and backward passes. Moreover, it would be beneficial to be able to mechanically drag, move, and extend individual activities in bar charts with logic links to directly observe their impact on the remainder of the schedule.

5 CONCLUSION

Creating crystal clear schedules with layouts that lend themselves well to the characteristics and requirements of the given projects is the first major step towards good use of the different existing scheduling techniques. Thinking of schedules in a more mechanistic way may assist with more intuitively understanding their structure, flow of operations, possible bottlenecks, and effect of disturbances in the system e.g. through delays or resource shortfalls. Unfortunately, project managers nowadays still much rely on simple network schedules and bar charts, potentially without at least resource-loading them, to organize their processes, make their decisions, and track their progress regularly and frequently. Ongoing research at The Catholic University of America currently investigates concepts to provide clarity in an automated manner for different types of schedules through use of network algorithms.

This paper has contributed to this endeavor by raising awareness of scheduling software default graphics settings that may obscure or distort actual schedule time and logic information. At the same time, careful review and graphical layout of the scheduling documents is encouraged so the communicating their full contents to other users is facilitated. Construction managers need to treat CPM scheduling software as that what it is – simply a tool, which always needs a trained hand and a creative mind for its correct application.

ACKNOWLEDGMENTS

The author would like to thank Dr. Julio C. Martínez of Virginia Polytechnic Institute and State University for helpful discussions on the topic of this paper.

REFERENCES

- Basu, A. 1990. A mechanical model for CPM scheduling calculations. *Transactions of the American Association of Cost Engineers Annual Meeting*: H.5.1–H.5.6.
- Basu, A. 2003. CPM scheduling in construction: A case study. *Transactions of the American Association of Cost Engineers Annual Meeting*: PS.04.1–PS.04.6.

- Melin, J. W., Whiteaker, B. 1981. Fencing a bar chart. *Journal of the Construction Division, Proceedings of the American Society of Civil Engineers* 107(CO3): 497–507.
- Tufte, E. R. 2001. *The visual display of quantitative information*. 2nd ed. Cheshire, Connecticut: Graphics Press.
- Wainer, H. 1984. How to display data badly. *The American Statistician* 38(2): 137–147.
- Winter, R. M. 1997. Management of CPM schedule submission. [online] Available via www.ronwinterconsulting.com/rasubmit.htm [accessed March 1, 2005].
- Winter, R. M. 2004. Longest path value (to the rescue). Presented at the *Project Management Institute College of Scheduling 2004 Annual Conference*, April 26, 2004, Montréal, Québec, Canada. [online] Available via www.ronwinterconsulting.com/Longest%20Path%20Value.pdf [accessed March 1, 2005].

AUTHOR BIOGRAPHY

GUNNAR LUCKO is an assistant professor and director of the Construction Engineering and Management program in the Department of Civil Engineering at The Catholic University of America (CUA). His research interests include construction operations analysis and optimization, equipment economics, and constructability analysis. He has studied statistical equipment valuation models and has participated in research for the Construction Industry Institute. He is a member of ASCE. His e-mail address is lucko@cua.edu and his web address is <http://engineering.cua.edu/faculty/profiles/lucko.cfm>.