

Graphical Planning Method®
Dr. Gui Ponce de Leon, PE, PMP, LEED AP
PMA Consultants LLC

Keywords: Active Planning Surface, Logic Diagramming Method, LDM, GPM, Objectbase, Collaboration

Abstract

The Graphical Planning Method (GPM®) transforms conventional Critical Path Method (CPM) planning and scheduling by making it an engaging, interactive, real-time process that results in a hands-on, planning-dominated experience for stakeholders. GPM allows project managers, superintendents, key subcontractors and other stakeholders to collaboratively, in one session, network a project by graphically positioning activities on an “active planning surface,” using a variety of simple and intuitive logic ties to convey activity relationships. GPM allows forward and backward planning, while, through its network graph algorithms, continuously refreshing floats. This results in creation of a network schedule in the shortest possible time. GPM allows floats to be realistically apportioned so delays to early activities do not consume floats available for downstream activities. If resources are associated with activities, GPM continuously displays the evolving resource profiles. Combining precedence and arrow diagramming into a new diagramming paradigm, GPM emphasizes the planning process for stakeholders charged with delivering the project but who are not professional schedulers [1].

GPM serves to interactively illustrate what impact activity-sequence variability has on overall project schedule performance. As a decision support tool, GPM depicts in real time how the project should be sequenced in light of means and methods, phasing, milestones and other such constraints. GPM fundamentally alters CPM practice by allowing evolving dates, floats, resource limits, crash-cost curves and risk to impact activity definition and logic. GPM is an apt preceding step to look-ahead scheduling and other techniques promoted for work-flow planning.

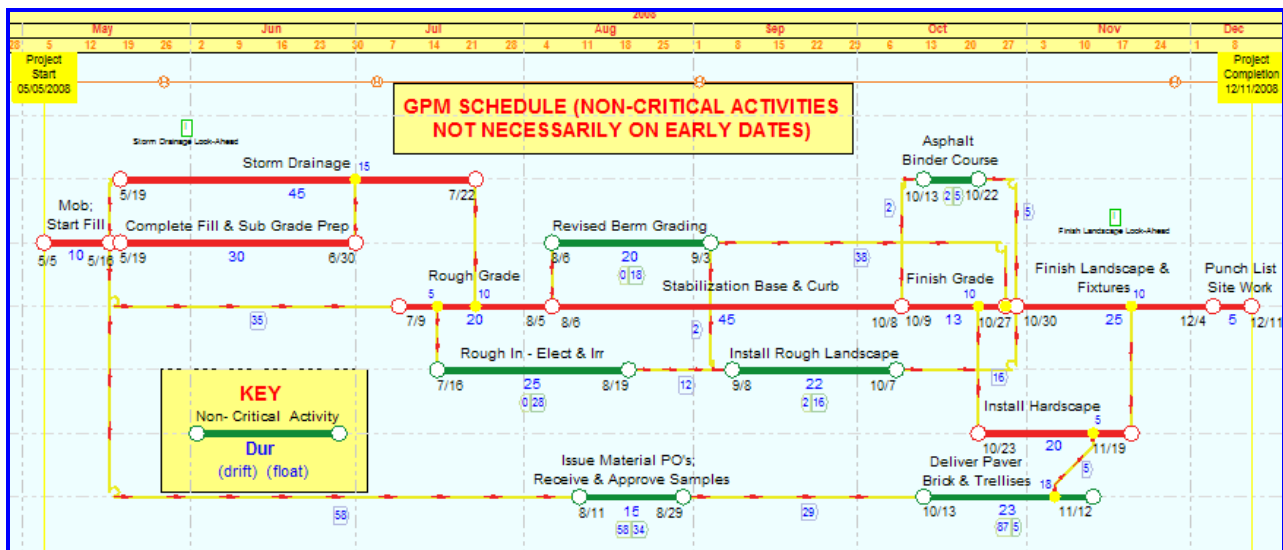
GPM® Technology

GPM is a networking technique that offers the simplest possible scheme of thought to create and optimize a project schedule in the shortest possible time. Innovative ideas are as follows:

- 1) GPM relies on interactive graphics that enable the visual display of planning objects (depicting the evolving schedule) as an intrinsic part of planning. The technology is comprised of an objectbase graphical user interface, which obviates the need for databases and scheduling engines to process the database, as all CPM-based applications do in a succession of batch modes;
- 2) The enabling network diagramming breakthrough is the Logic Diagramming Method (LDM); LDM, while retaining the graphical simplicity of a user-controlled arrow diagram, offers the relationship flexibility heretofore only associated with Precedence Diagramming Method (PDM), without requiring practitioners to go through the arduous process of learning the complexities of PDM relationships;
- 3) Planning on a time-scaled calendar as opposed to schematically is more efficient and allows planning to evolve spontaneously rather than sequentially, and further, planning may switch from working forward from release dates to backwards from target dates or milestones and vice versa;
- 4) All activity floats originate with the relationships or logic ties, therefore, activity floats are a function of relationship leeway, or *link gap* in GPM terminology; such a function is algorithmic [2];
- 5) Floats can be realistically apportioned by not letting activities slip beyond assigned fixed events;
- 6) If resources are associated with the activities, resource-limited activity dates and floats are solidified as the plan evolves as opposed to having to iterate using (albeit rarely used) global resource leveling;
- 7) Owing to its foundation on network graph algorithms, when cost effectively crashing and extending a schedule, GPM is capable of spontaneous healing of corrupted (bad logic) plans based on mathematical rules without having to call on a scheduling engine; and
- 8) As the evolving schedule nears full development, stakeholders can resolve resource constraints, study optimal schedule compression/extension scenarios and otherwise rework the pre-optimized plan visually without relying on black-box calculations performed by CPM software.

The Logic Diagramming Method (LDM) – A synopsis

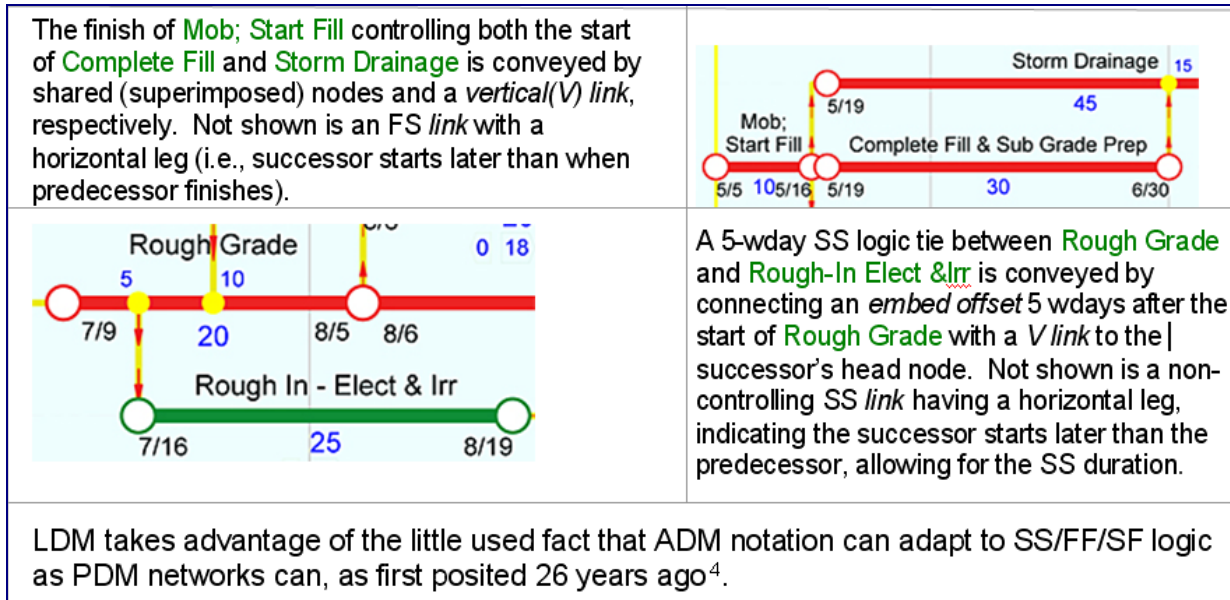
ADM (Arrow Diagramming Method) and PDM (Precedence Diagramming Method) are well established in CPM practice. ADM denotes activities as arrows between start and finish nodes and connects dependent activities at their nodes, which allows only finish-to-start (FS) logic. PDM places activities on boxes and uses links to connect the finish or start of an activity to the start or finish of a successor, which allows four types of dependencies. LDM is a new method of constructing project networks that modifies and extends ADM to permit the four types of PDM logic. In LDM, activities are time-scaled, ADM-like lines, which obviates the need for activity arrowheads [3].



GPM Sample Schedule Using LDM Notation

As illustrated in the above example, LDM activity notation resembles ADM notation, albeit on a time scale. Logic ties have multiple arrowheads. Start-to-start (SS), finish-to-finish (FF) and start-to-finish (SF) logic is accepted through *embedded nodes*, intermediate of, or right on, activity start and finish nodes. Driving relationships are conveyed by a common node (FS only) or, owing to the time scale, a *vertical link*. Non-controlling *links* include a horizontal segment or leg, which denotes the *gap* (calculating the amount of leeway existing) in the relationship.

In LDM notation, relationships are viewed as connecting a) two nodes, b) an *embedded node* (*embed*) and a node, or c) two *embeds*, making relationship types no longer relevant. For instance, depending on activity positioning, FS and SS relationships will graphically appear in LDM as follows:



Finish-to-Start (FS) and Start-to-Start (SS) Logic on LDM Notation

Planning and Scheduling Consolidate Into One and the Same Process

With GPM, once an activity is *positioned* or *re-positioned* on *planned* dates, *floats* in the evolving plan instantaneously calculate without the necessity of a conventional CPM backward pass. In this respect, GPM is a networking process of one-step flow, instead of a flow consisting of batches [5].

Allowing for individual preferences [6], batching in CPM may break down as follows:

Common Steps Followed when Scheduling a Project	Common to CPM Practice		GPM
Scoping the project – what makes up the completed project	BATCH 1	PLANNING	PLANNING (ONE-STEP FLOW)
Scoping the approach to the project, phases and milestones			
Scoping the phases into activities that have to be performed			
Sequencing the activities, that is, how do they interconnect			
Calculating activity earliest dates (when <i>can</i> be performed)	BATCH 2	SCHEDULING (STILL PLANNING FOR SOME)	
Calculating activity latest dates (when <i>must</i> be performed)			
Determining activity total floats (first come, first served)			
Re-working durations and sequences until acceptable	BATCH 3	SCHEDULING	
Establishing realistic early dates that level the work force	BATCH 4	SCHEDULING	
Having to rework software-determined realistic dates	BATCH 5	SCHEDULING	N/A
Searching for an optimal plan in light of time dependent costs	BATCH 6	SCHEDULING	
Working the plan, i.e., proceeding to look-ahead scheduling	BATCH 7	SCHEDULING	SCHEDULING

Batched CPM Practice v Non-batched GPM Protocol

Simultaneity of activity definition and sequencing, *planned* dates determination and *float* calculation means that GPM planning is never batched before scheduling begins; planning runs coterminous with scheduling. Determination of *planned* dates reverts to a user-driven, planning function. Thus, it is posited that GPM is a planning-dominated scheme of thought, the result of which is a perfected, optimized network schedule, including realistic dates and *floats*. This is a departure from the current planning and scheduling paradigm as echoed by AACE International [7] and others.

Planning on the Calendar v. Schematically

In GPM, *planned* start/finish dates flow naturally when activities are assigned durations and sequenced on the time-scaled calendar, obsolescing forward pass calculations. Knowing *planned* dates as activities are being sequenced allows for re-planning to flow spontaneously. This is not possible in the realm of schematic activity definition and sequencing that takes place in the abstract.

With the calendar in full view, planning, at any time, can move on to a completely unrelated set of activities, logic and milestones, and then eventually resume on prior activities. In situations where many activities converge with few successors, planning may switch to a *pull* mode, with predecessor

activities and relationships developed backwards from already defined successor activities. As positioned and in early implementation cases, GPM was envisioned and has proven to allow activity definition, sequencing and dating to proceed in a push/pull fashion [8].

Stakeholder-driven v. Software-calculated Dates

GPM provides a scheme of thought that, except for the calculation of *gaps, floats and drifts* (Glossary), concedes to human judgment where most appropriate. When an activity is first added to the evolving plan and connected, it may be *positioned* on the early dates the CPM forward pass would yield (“default” case). Once connected to predecessors and successors, however, stakeholders get other opportunities to *re-position* the activity and/or rethink its duration and even the degree of overlapping so as to achieve a more optimal scenario.

Rather than relying on CPM software heuristics that very few can use or comprehend, provided resources and time/cost trade-off data are associated with the activities as they are *positioned* or *re-positioned*. GPM lets users make their own choices as to which activities to move to earlier or later time frames, crash or extend to improve manpower loading and optimize time-dependent costs. GPM suits itself to the users’ own heuristics and thought process developed through years of collective experience doing so with or without CPM [6].

Float Management

GPM allows activities to be connected to fixed events or *benchmarks*. This allows *floats* to be apportioned by not positioning activities after or before assigned *benchmarks*. When using *push* planning, leeway materializes between an activity and its *benchmark* anytime the relevant *planned* date of the activity preceding the *benchmark* is earlier than the *benchmark* date. This may appeal to

Lean Construction proponents and those aware that the CPM concept of total float should attach a disclaimer: scheduler beware! With some exceptions [9] [10], few practitioners are equipped to adjust conventional CPM floats for the “total float paradox:” total float accumulates from schedule completion backwards; if an earlier activity consumes any of its total float, the float available for downstream related activities is correspondingly reduced.

Conclusion

Because GPM facilitates collaboration, it provides a renewed opportunity for communication and joint planning of a project with key subcontractors as well as the prime, a practice that over the past thirty years has taken a back seat to schedule manipulation by specialists not directly responsible for delivering the project. In contrast to past practice, GPM emphasizes experienced human judgment and collaboration in project planning instead of machine calculations. Far too often, past scheduling practice has relied on machines and software to produce CPM schedules that, although technically error-free, were not well coordinated with all stakeholders, did not correlate well with project reality and were therefore unworkable and often ignored during project execution.

Glossary

Activity – A *positioned GPM* object with assigned duration representing a discrete part of the work, with or without assigned resources and cost attributes, or a zero-resource restraint to another *Activity*.

Benchmark – A GPM event object designating a deadline or used to “cut” floats (drifts), if floats (drifts) are allocated to the phase defined by the benchmark v. accumulating from completion (to project start).

Drift– GPM-calculated *Activity* and *Milestone* attribute, which measures by how many days an *Activity* or *Milestone* may backslide to an earlier *Position* without forcing an earlier project start.

Embedded Node (Embed) – An event intermediate of, or right on, the start and finish nodes of an *Activity*, through which the *Activity* is connected to a successor start node, from a predecessor finish node, to/from an *embed* of another *Activity*, or to a *Benchmark*.

Float – GPM-calculated attribute, which measures by how many days an *Activity* or *Milestone* may slip beyond its *Position* without extending overall schedule completion. An analogous concept to Total Float, except it is measured with respect to *Planned Dates* as opposed to “early dates.”

Gap – GPM-calculated *Relationship* attribute measuring leeway by how many days the predecessor may slip beyond its *Position* without delaying the successor’s *Position (Activity or Milestone)*.

Milestone – A GPM event dating a key point in the schedule, which, if constrained, distributes the project duration to preset stages, and, if unconstrained, reflects the forecasted date for the milestone.

Objectbase – A collection of rule-encapsulating computer objects that, in response to external events, inherently interact via message passing to drive a visual model” - such obviating the need for in-line calculating engines waiting for their calculating turn. By way of comparison, database is a systematically arranged collection of computer data, structured so that it can be automatically retrieved or manipulated with such retrieval or manipulation carried out by calculating engines that competently apply the pertinent rules, mathematical or otherwise. GPM's visual, real-time orientation favors objectbase v database software applications.

Offsets – Analogous to PDM lead/lags. An FS *Offset* denotes the interval between the finish of the predecessor and start of the successor; an SS *Offset* denotes the interval between the start of the predecessor and start of the successor; and so forth for FF and SF relationships.

Relationship – A time-specific interconnection between two dependent GPM objects. In *LDM*, 7 possible types of *Relationships* with *Offsets* are: finish-to-start; embed-to-start; finish-to-embed; embed-to-embed; finish-to-benchmark; start-to-benchmark; and embed-to-benchmark. PDM allows finish-to-start; start-to-start; finish-to-finish; and start-to-finish. ADM allows only finish-to-start.

References

[1] Ponce de Leon, G. (2008). *Graphical Planning Method (A New Network-based*

Planning/Scheduling Paradigm. Presented at the PMICOS 5th Annual Conference, Chicago, IL

[2] Ponce de Leon, G. (1972). *Extensions to the Solutions of Deterministic and Probabilistic Project Network Models*. Ph.D. Thesis. Civil Engineering Department, College of Engineering, Univ. of Michigan, MI.

[3] Ponce de Leon, G. (2008). *Project Planning using Logic Diagramming Method*. Presented at the AACE International 52nd Annual Conference, Toronto, CA.

“LDM, with its *embedded node* construct and time-scaled v. schematic diagramming synergizes what PDM and ADM individually provide. LDM’s modeling of PDM logic through embeds treats PDM leads/lags as elapsed days of work related to the predecessor and successor, respectively.”

[4] Ponce de Leon, G. (1983). Overlapping Arrow Networks. *Stratagem, A Project Controls Journal* 1(2), 1-2.

Arrow networks can adapt to overlapping logic as well as precedence networks can by adding dummies for each relationship in an arrow diagram even if not required by dummy rules.

[5] Womack, J.P. & Jones, D.T. (2003). *Lean Thinking: Banish Waste and Create Wealth in Your Corporation*. New York, NY: Simon & Schuster, Inc.

In Lean Thinking versus Muda, the authors state “We are all born into a mental world of “functions” and “departments,” a commonsense conviction that activities are to be grouped by type so they can be performed more efficiently and managed more easily...Batches, as it turns out always means long waits...”

[6] Clough, R.H., Sears, G.A. & Sears, S.K. (2000). *Construction Project Management*. (4th Ed). New York, NY: John Wiley and Sons, Inc.

“Construction planning may be said to consist of 5 steps: (1) a determination of the general approach to the project; (2) breaking down the project into job steps or “activities” that must be performed to construct the project; (3) ascertainment of the sequential relationships among these activities; (4) the graphic presentation of this planning information in the form of a network, and (5) endorsement by the project team.”

[7] AACE International (2007). *Draft Planning & Scheduling Professional Certification Study Guide*. AACE, Morgantown, WV: AACE.

- **Planning** – Identification of project objectives and the orderly activities necessary to complete the project (the thinking part) and not to be confused with scheduling; the process by which the duration of the project task is applied to the plan. It involves answering the questions: 1) What must be done in the future to reach the project objective; 2) How it will be done; 3) Who will do it; 4) When it will be done?
- **Scheduling** – A description of when each activity in a project can be accomplished and must be finished so as to be completed timely.

[8] Koskela, L. (2004). Moving on – Beyond Lean Thinking. *Lean Construction Journal*, 9(1) 24-37.

It is noted that a production control system can be a mixed push-pull system. And that either push and pull method may be appropriate depending on the stage in question. This is relevant to construction planning in that deliveries allow the release of work, which suggests that work dependent on deliveries may best be sequenced using “push techniques”.

[9] Ponce de Leon, G. (1983) Activity Float Entitlement. *Stratagem, A Project Controls Journal*, 1(3).

“Together, these activities make up the early schedule tree. This network allows identification of all “chains,” that is, those sequences of activities of identical total float...By reviewing the float along these chains, one can see the inequities inherent in conventional float concepts.”

[10] Woolf, Murray (2007). *Faster Construction Projects with CPM Scheduling*. NY, NY: McGraw-Hill.