Links, Lags and Ladders

- the subtleties of overlapping tasks -

Logic in a Precedence Network

Precedence diagrams use boxes to represent the basic network elements - the task (or activity). Tasks have a duration (the period of time required to perform the work they represent), a description, and may have other data attached to them. The other key element of precedence networks is the dependency (or link), which defines the logical relationship between the tasks. A link is shown in a precedence network diagram as a line.

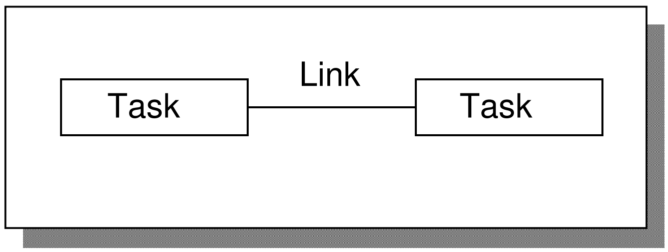


Figure 1 – Tasks and Links

Tasks are identified by a task identifier - for example, A1, A2, A3. Links are usually identified by their preceding task identifier and their succeeding task identifier.

The other element that should be included in every schedule is Milestones. Milestones are 'zero duration' events that mark significant points in the schedule such as its start and finish and are connected to other tasks and milestones with links.

Logic describes the flow of work

The relationships between the tasks define the flow of work through the project. The objective is to organize the tasks into a logical sequence agreed to by the project team. Only real logic should be used to construct the logic diagram (or network) using Finish-to-Start relationships where possible. Real logic can be:

____

- Dictated by the intrinsic nature of the work
- Mandated by the contract
- External to the project representing either a deliverable required for the work to continue, or something the project has to deliver to a third party, or
- A sequence of work that is an express intention of the project team

The first two options above are mandatory logic; the third is an 'external dependency', the last is discretionary logic; but they are all 'real'. Artificial logic inserted to fix a problem should be discouraged as it distorts the schedule and can have unintended consequences as the schedule changes during the life of the project.

Dependency Management

External dependencies require a different management approach to internal logic (discussed in the balance of this paper).

'Outgoing' dependencies represent requirements of other projects or an interim deliverable to the client. These are either a contractual requirement which represents a constraint that has to be achieved, or an obligation to assist the overall running of the organization's total project delivery effort. The receiver of the outgoing link is a stakeholder of the project whose needs are important and should be met wherever possible.

'Incoming' dependencies are a risk! They represent requirements the project needs to complete its work but the project team does not control the delivery process and the risk needs managing.

Managing external dependencies requires a significant focus, including:

- The identification of the dependencies (at an appropriate level of detail)
- Mapping the dependencies into the schedule (we recommend highlighting each dependency with a milestone)
- Determining the way the dependencies will be technically mapped between projects (there are various software options – fully automated linking is not recommended)
- Determining how the progress on delivering incoming dependencies will be monitored and variances managed
- Recording key risks in the risk register, and
- The on-going management of the dependencies as work progresses.

External dependencies are similar to the schedule start and finish date in terms of framing the overall project plan.

Developing 'internal' Logic

To determine what constitutes a logical relationship within the schedule the key questions to ask are:

- What has to be completed to allow this activity to start?
- What cannot start until this activity is completed?
- What can happen at the same time as this activity?

The resulting logic is a 'road map' showing the sequence of work from the beginning to the end of the project.

When this process is complete, every task and milestone should be connected from its start to at least one predecessor and can trace its logical predecessors to the Start Milestone and from its finish to at least one successor and can trace its logical successors to the Finish Milestone[1].

Summary activities

Summary activities can be created in a variety of ways (depending on the tool being used) and are useful for reporting purposes and also for carrying certain types of cost and resource information. However, for effective schedule management, summary activities and 'Hammocks' should be a 'roll up' of the detail information in the schedule - they should not control the schedule. Therefore, good practice dictates the summary activities should not be logically linked (the links should be at the detail level).

Links in a Precedence Network

As already mentioned, links dictate the flow of work through the project. There are four basic types of link: Finish-to-Start (FS), Finish-to-Finish (FF), Start-to-Start (SS) and Start-to-Finish (SF). Of the four types of link, FS links are the most common and SF links are rarely used. However, using any type of link other than FS can produce unexpected results during schedule analysis as they have not been consistently implemented by project management software developers (see *Logical Inconsistencies* below).

Finish-to-Start Links

The normal type of link is a Finish-to-Start link (FS). With this type of link, the succeeding task cannot start until after the finish of the preceding task.

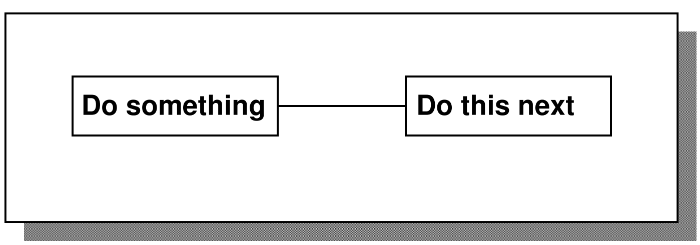


Figure 2 - Finish to Start Link

If a lag time is specified on the link (say 3 days), the succeeding task cannot start until three days after the finish of the preceding task.

[1] See: **Dynamic Scheduling**: https://mosaicprojects.com.au/PDF-Gen/dynamic_scheduling.pdf

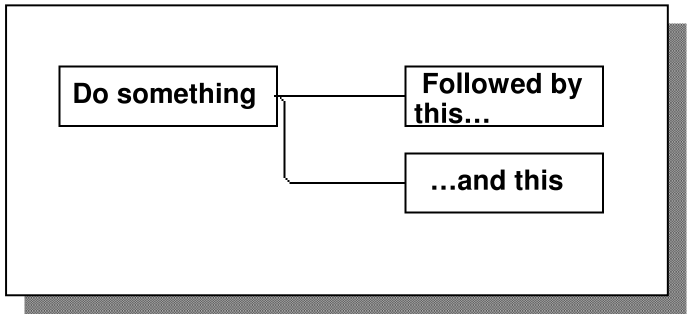


Figure 3 – Succeeding Tasks

Links work independently. In Figure 3, neither of the following tasks can start until after the leading task is finished BUT they do not have to start at the same time and they do not have to proceed together.

Finish-to-Finish Links

Finish-to-Finish links (FF) constrain the completion of a task. The completion of the succeeding task is delayed until after the completion of the preceding task. If a lag is nominated (say three days), the finish of the succeeding task is delayed until three days after the finish of the preceding task.

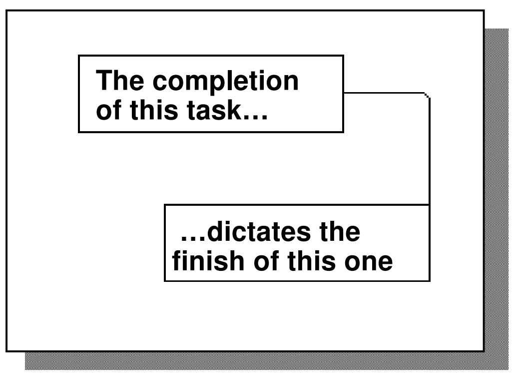


Figure 4 – Finish-to-Finish Link

This type of dependency primarily controls the finish of tasks (not the start). A typical example would be writing and editing a book. The editor does not have to wait until the writing is finished to start the editing process; editing could start as soon as the first chapter is finished. BUT, it is impossible to finish editing until after the writing is complete. The editor may require a week to complete the editing once the book is finished and this is represented by creating a Finish-to-Finish link with a lag of 5 days.

Start-to-Start Links

Start-to-Start links (SS) constrain the start of a task. The start of the succeeding task is delayed until after the start of the preceding task. If a lag is nominated (say three days), the start of the succeeding task is delayed until three days after the start of the preceding task.

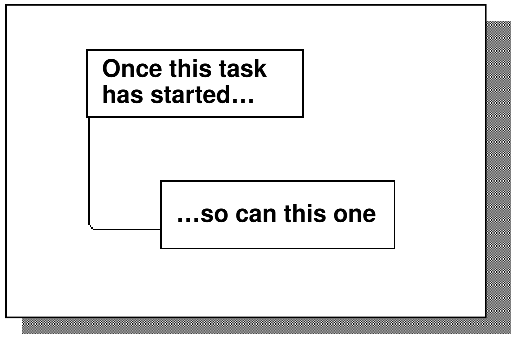


Figure 5 – Start-to-Start Link

This type of dependency primarily controls the start of tasks (not the finish). Staying with the writing and editing of a book, it is also impossible for the editor to start editing until some of the writing is complete (maybe the first chapter). The author may require two weeks to format the overall plan for the book and write the first chapter. This is represented by creating a Start-to-Start link with a lag of 10 days.

If you need to control both the start and the finish of the relationship between two tasks (as would be the case with writing and editing), it is best to insert both links between the tasks (SS and FF). If this is not possible (some software will only allow one link), then you must decide which link is most important (see: *Managing the Overlap* below).

Start-to-Finish Links

Start-to-Finish links (SF) constrain the finish of a task based on the predecessor starting. The finish of the succeeding task is delayed until after the start of the preceding task. If a lag is nominated (say three days), the finish of the succeeding task is delayed until three days after the start of the preceding task.

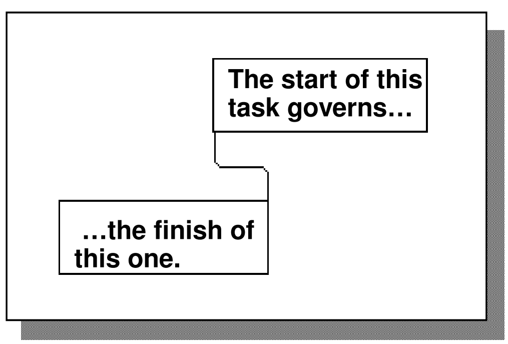


Figure 6 Start-to-Finish Link

This type of link is typically used to control the change-over between two processes, if a business is changing from a security system that uses key cards for access to one that uses bio-metrics, the use of the key card system cannot finish until after the start of the bio-metric system. If both systems are required to run in parallel for a time, a lag is added to the S-F link.

Loops and Open Ends

Loops

The concept of a loop is almost impossible to build in a single schedule – graphical software simply refuses to allow one to be created. However, in the days when data was fed into a computer one element at a time using punch cards and the like, it was fairly easy to set up a scenario where the link information stated:

- Activity A is followed by Activity B
- Activity B is followed by Activity C
- Activity C is followed by Activity D
- Activity D is followed by Activity A

Computers would hit this logical 'loop' and keep spinning through the sequence A-B-C-D-A……. The only solution was to switch off the program and sort out the logic.

This problem can still occur if data is batch input (although modern software has 'loop detectors' built in and advises of the issue requiring the loop to be corrected before analysis can take place).

Loops can also occur (and are much harder to detect) if external links are used to connect different programs together in an 'enterprise' situation – Your schedule has activities A-B-C with an external link from C to Activity X in their schedule; their schedule has activities X-Y-Z with and external link from Z to Activity A in your schedule. Generally, the only way you find the issue is the completion date keeps jumping back each time 'your' schedule is analyzed following an analysis in the 'other' schedule.

Open Ends or 'Dangles'

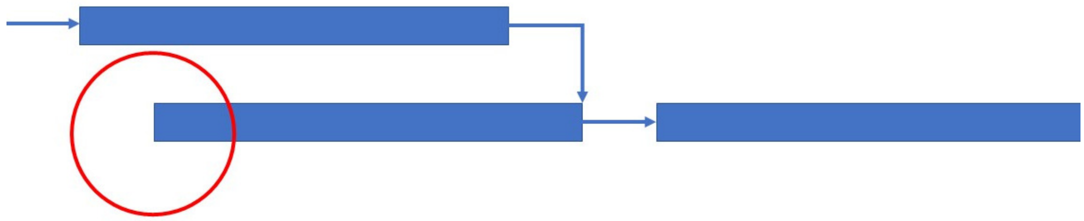
With the exception of a 'Start Milestone' and an 'End Milestone', the scheduling standards require every activity to have link connecting a predecessor to its start and a link connecting is finish to a successor. The consequence is every activity can trace a logical path from the start milestone to its start and from its completion to the end milestone[2]. An 'open end' or 'dangle' occurs when one of these requirements is not met.

The complete absence of a predecessor or successor is an obvious error and easy to identify and correct. However, it is possible to create dangles when an activity is connected into the network with both predecessor and successor links. The most common issues are:

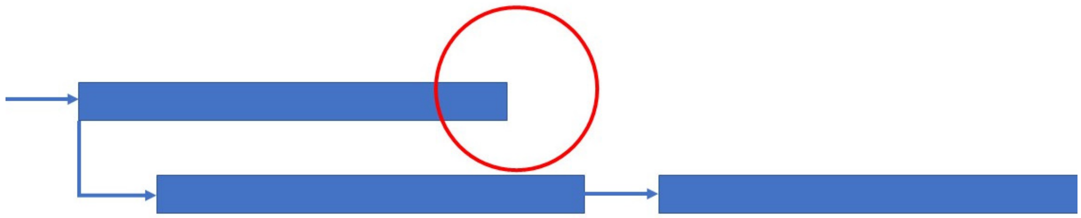
- If the only predecessor to an activity is connected using a finish-to-finish link there is no predecessor connected to the start and therefore a 'start dangle' exists. The start of the activity has

[2] This is a fundamental requirement for a ***Dynamic Schedule***: https://mosaicprojects.com.au/PDF-Gen/dynamic_scheduling.pdf

to be presumed based on the logical connection to its finish.



- If the only successor to an activity is connected using a start-to-start link there is no successor connected to the end and therefore an 'end dangle' exists. The completion of the activity has to be presumed based on the logical connection from its start and once the start link has triggered the start of the successor there is no logical constraint on when the activity should finish.



- If the only successor to an activity is connected using a finish-to-start link with a 'lead' (or negative lag) applied to the link once the 'lead' has triggered the start of the successor there is no logical constraint on when the last part of the activity should finish (see below).
- Other link types such as progressive feed and percentage overlaps (discussed below) can also create 'dangles' usually in the last part of an activity – whether a 'dangle' exists or not depends on the algorithms used in the scheduling methodology and its implementation in the software being used.

Ensuring the logic is complete and 'sensible' is a key quality assurance step in the process of developing a competent schedule.

Leads and Lags

As described above, a 'positive lag' has the effect of delaying the succeeding task by the number of time units specified. Negative lags (or 'leads') have the effect of accelerating the succeeding task by the number of time units specified. Consequently, if the lag value is specified as a negative number, it has the effect of overlapping the tasks. A lag of - 3 days on a F-S link would mean the succeeding task can start 3 days before the end of the preceding task (ref: Fig. 7).

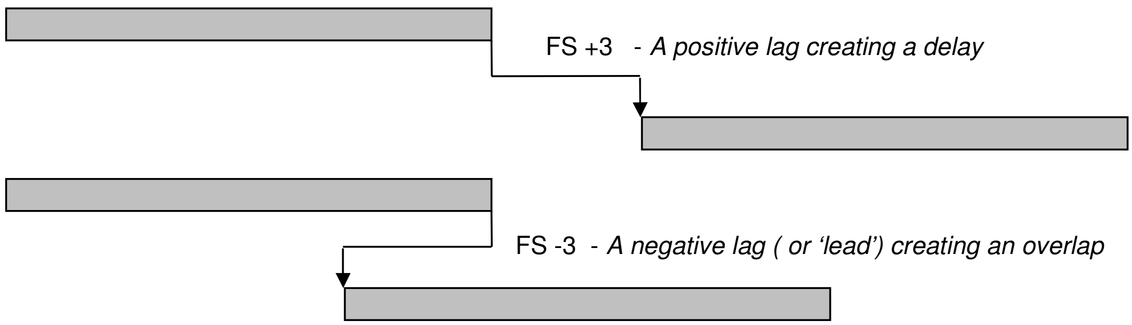


Figure 7 Leads and Lags

Adjusting the degree of overlap between activities (or groups of activities) is one way of accelerating the planned work and reducing the overall duration of the project (Fast Tracking)[3]. What is important to remember is making an adjustment in the schedule is much easier that it is in the 'real world' - ultimately for the schedule to be of any use it has to be both realistic and achievable.

Lags should not replace work. Even where work is to be performed by others, this work should be included as a task. For example, if the contact allows one week for the review of a drawing by the client; do not insert a lag of 5 days on the link between the task for creating the drawing and the task for using the drawing (both your work). Rather, insert a 5-day task for the client review; this task can then be coded and reported upon during status updates of the schedule[4] and any delays properly attributed to the responsible party.

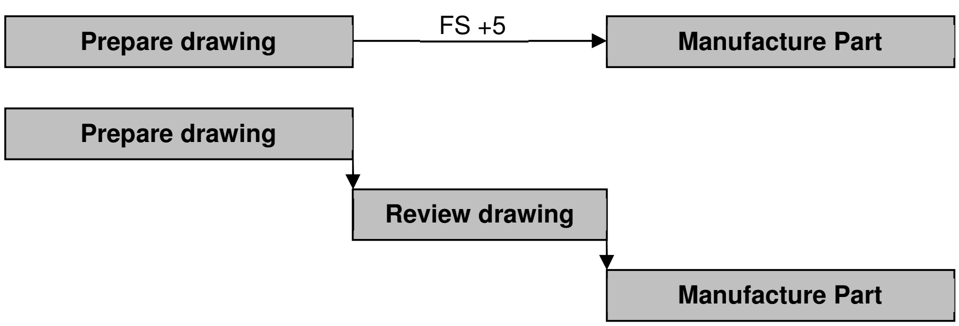


Figure 9 - Lags should not replace logic

If the time between the activities is needed for a purpose, but no work is happening (eg, concrete curing time or paint drying time) a FS lag is appropriate and the 'space' has a purpose. However, Lags should not be used simply to create a space between two activities 'for convenience' or to make the schedule look correct. These *'leaps of logic[5]'* bypass true network logic by linking tasks with inherent gaps in time

3 For more on ***schedule compression*** see: https://mosaicprojects.com.au/WhitePapers/WP1059_Schedule_Compression.pdf

4 See: ***A Guide to Scheduling Good Practice***: https://mosaicprojects.com.au/PDF-Gen/Good_Scheduling_Practice.pdf

5 Term developed by Jim Peter and Kelvin Murray to describe this effect.

between the activities and can be misleading and may cause computational errors when used; the effect is similar to putting artificial constraints in the schedule and should be discouraged.

Leads should be used with care. Negative lags (or leads) are allowed in some software packages and are a legitimate 'tool' in the schedulers tool-kit to create an overlap between two activities but need to be used with care. From a logical perspective a negative lag is difficult to justify and its use is discouraged or prohibited by many scheduling standards and guidelines. In most circumstances the combination of SS and FF lags can achieve a more sensible overlapping of activities. However, because a number of limited tools only allow a single link between activities, the concept of a 'Lead' (or negative lag) is retained in this paper and other authorities such as the *PMBOK® Guide*.

As shown in Fig. 7 above, a lead defines the start of an activity by referencing the completion of its predecessor, however the successor starts before the predecessor is complete. This arrangement is very useful for scheduling handovers and the like where the people involved in the predecessor need to transfer knowledge to the people who will be working on the successor activity but has significant logical inconsistencies. The major issue is that once the successor starts, there is no logical dependency controlling the completion of the 'last bit' of the successor – in effect this creates an 'open end'.

Managing the Overlap

Where inserting an additional task is not appropriate and the gap is 'real', the nature of the gap needs to be clearly understood[6]: Why is this lag needed?

- Does the time represent an imposed delay to create a sensible flow of work allowing the leading task to clear sufficient work space for the succeeding task to commence within?
- Does the time represent administrative works needed to prepare for the succeeding task?
- Does the time represent a productive work segment (Ref: Fig. 10 & 11) where a certain amount of work has to be completed on Task A before Task B can start to use the handed over work?

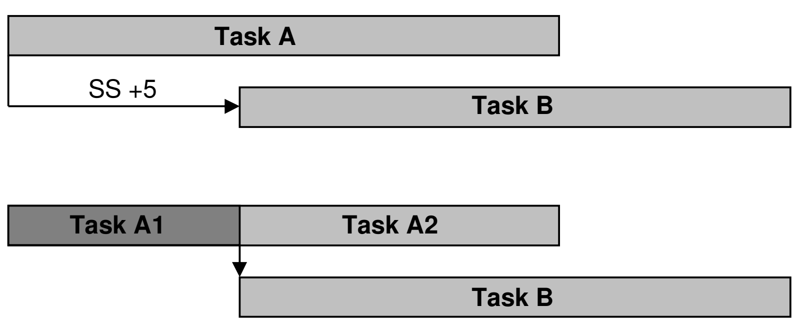


Figure 10 SS Link = Productive work segment

[6] See: ***Faster Construction Projects with CPM Scheduling***, 'Anatomy of a relationship' page 177.

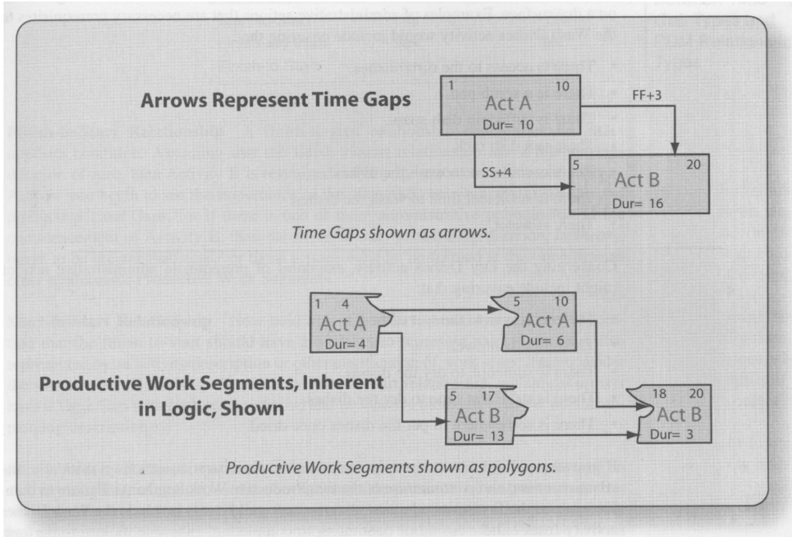


Figure 11 - Extract from Woolf's Book[7]

Understanding the nature of the relationship is critical to effectively managing the schedule; anecdotal evidence suggests most of the minor delays that are the responsibility of the project team (ie the contractor) occur in the gaps between tasks represented by lags. In aggregate these delays can have a major impact on the momentum of the project and cause delays to completion.

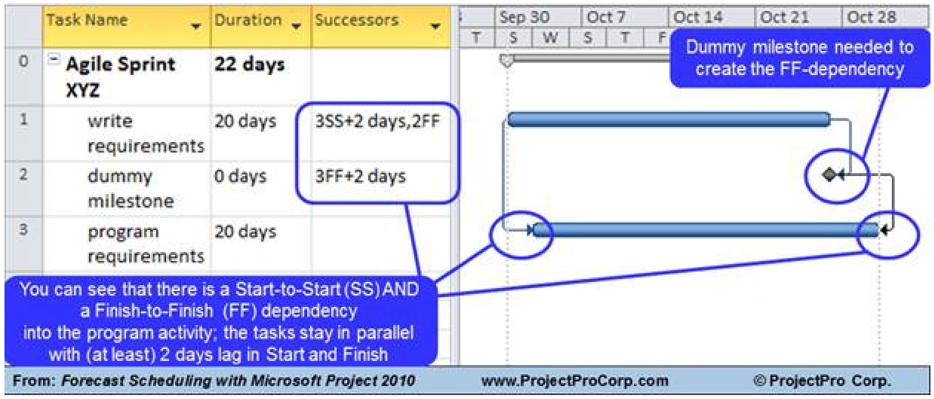
Where only one link is used the next question is does the remaining part of Task A have any influence on Task B? In the case depicted in Fig. 10, there is a high probability that all of the work in Task A has to be completed to allow Task B to finish, but this is not necessarily the case. However, if there is a need for Task A to continue to feed work to Task B our strong recommendation is to either:

- Set the link type to 'progressive feed'; a number of tools have this feature. Progressive feed only allows B to progress proportionally to A.
- Use both a SS and a FF link to at least constrain the start and finish of B in relation to A.

If your current tool is incapable of either and you want to develop useful schedules that produce predictable results during the progress of the works either, stick exclusively to Finish-to-Start links, buy a software tool that works, or add some additional logic to simulate the effect.

The problem with inserting dummy logic (as per the example below) is the tool cannot manage the dummy relationship and milestone – you have to do the work. Failing to remember the 'dummy milestone' will sooner or later cause an error in your updating.

[7] Woolf, M.B. (2007) *Faster Construction Projects with CPM Scheduling*. McGraw-Hill, New York.



Dummy logic is necessary in some unsophisticated tools.

Logical Inconsistencies

As previously mentioned, the use of links other than Finish-to-Start can cause unexpected problems. Fig. 12 represents the dry walling work on Level 5 of a high rise block of units (one complete floor):

- Task A is the erection of the framing. This 10 day activity involves 2 days to set out the walls and fix the head and floor tracks and 8 days to fix the rest of the studs and frames
- Task B is the in-wall services rough-in. This involves a total of 3 days work by electricians, plumbers and others to run their pipes and cables inside the wall ready to connect to fixtures and fittings at a later date. This task can start 4 days after Task A has started (this allows time for the framers to have installed around 25% of the studwork) but cannot finish until 1 day after all of the framing is installed. By its nature this work is intermittent requiring several short visits to the floor by each of the services trades.
- Task C is the fixing of the wall sheeting. This can start one day after the 'in-wall services rough-in' has started and needs 3 days to finish after the last of the services are installed in the wall. The three days allows sufficient time to fix the last sheets, finish setting the joints and on the final day complete the sanding of the joints. However, fixing, setting and sanding the wall sheeting will take 12 days overall. Progress on the wall sheeting is only partly dependent on the in-wall services because not every wall has services inside it and as long as the service trades have access to one side of the walls where there are internal services, the sheeting can be installed on the other. The sheeting also needs at least 3 days after the completion of the framing (Task A) before it can finish.

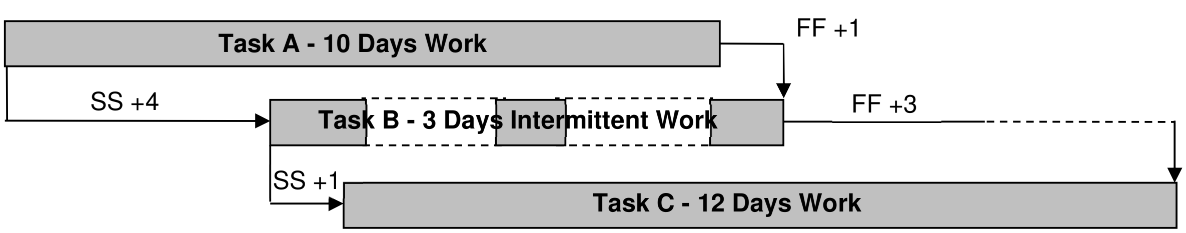


Figure 12 - Wall Framing Level 5

The situation in Figure 12 represents the optimum situation[8]. Task B starts 4 days after Task A allowing Task C to start one day later. Task B finishes 1 day after Task A allowing Task C to complete 12 days after it started. The overall duration of this work is 4 days at the start of Task A, plus 1 day at the start of Task B plus the full 12 days for task C equaling 17 day work[9].

The calculation of Float[10] in this situation is interesting! Only the first 4 days work of Task A are actually critical, and only the first day's work of Task B is critical. Looking at the completions, Task B can finish on Day 11 (10 days work on Task A plus one day to finish off Task B). However, Task B has a Finish-to-Finish relationship to Task C of FF+3. This means Task B does not have to finish until Day 14, which would still allow the 3 days (day 15, 16 and 17) needed to complete the wall sheeting. Given Task B can finish on Day 11, but its finish could be delayed until Day 14, and this delay will have no effect on any other work, arguably the completion of Task B has 3 days Free Float (but not the whole task). A similar conundrum exists with Task A; it can finish up to 3 days late and will only delay the finish of Task B which has 3 days float.

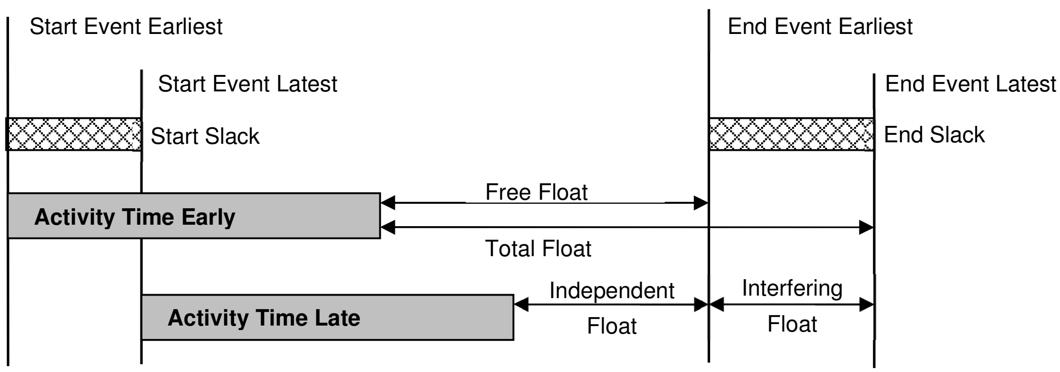


Figure 13 - Types of Float[7]

From the 1960s through to the 1980s, (and particularly with Activity-on-Arrow networks) float was dealt with in a far more sophisticated manner than today's simple calculation of Free Float and Total Float. The range of float options is set out in Fig. 13 and many of these ideas have been incorporated in the new scheduling methodology, RD-CPM™, the Relationship Diagramming variation of the Critical Path Method[11].

The calculations in a standard Precedence network should assess the situation at the start of the activity (the Start Event) and the completion of the activity (End Event). All of the above 'floats' have relevance in efficient resource levelling algorithms, unfortunately they are rarely considered[12].

Unfortunately, very few of today's software tools will resolve the situation in Fig. 12 satisfactorily. Most will resort to the solution in Figure 14; delaying Task B to comply with its finish link and schedule 'B' from Day 9 to Day 11. The consequence of this is to push the start of Task C to Day 10 and the end of the three tasks to

---

8 **Note**: It is possible to start the wall sheeting one day earlier (in parallel with starting the in-wall services) but this assumes the first wall available for sheeting do not have in-wall services – prudence suggests allowing the in-wall trades to get a start.

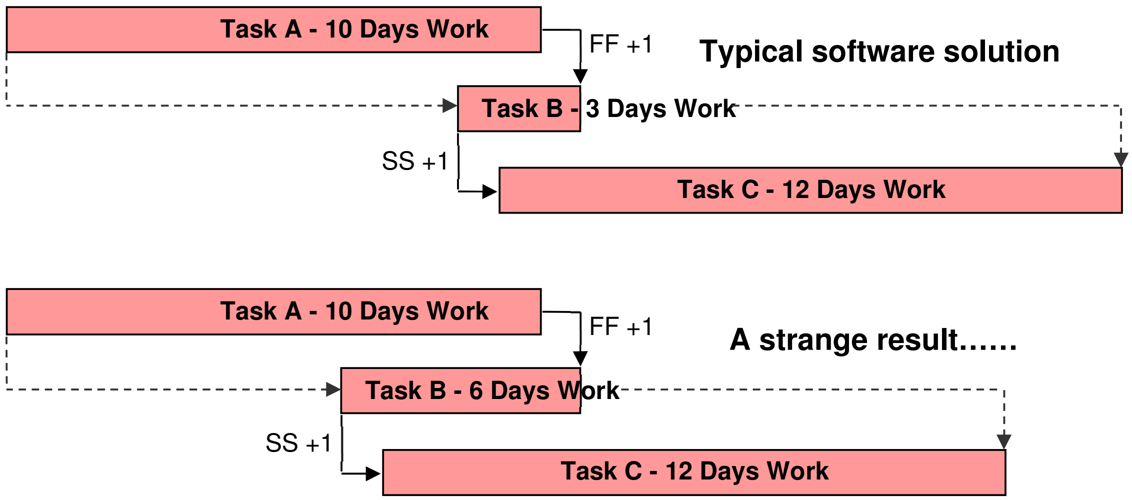
9 A full discussion on CPM calculations is in ***Basic CPM Calculations***, see: https://mosaicprojects.com.au/PDF-Gen/Schedule_Calculations.pdf

10 For more on ***Float*** see: https://mosaicprojects.com.au/PDF-Gen/Schedule_Float.pdf

11 For more on ***RD CPM™*** see: https://mosaicprojects.com.au/WhitePapers/WP1035_RD-CPM.pdf

12 For more on **Schedule Float** see: https://mosaicprojects.com.au/PDF-Gen/Schedule_Float.pdf

Day 21. This effect is described as 'lag drag'. Paradoxically, in this situation the whole of Task B is critical, but increasing the duration of Task B actually reduces the overall time for the three tasks to complete[13].



Increasing the duration of 'critical' Task B reduces the overall duration of the work!

Figure 14 - Some typical software induced problems

Ladders

The ladder technique was invented in the UK by ICL in the early1960s[14] (now Fujitsu), and gained wide acceptance in scheduling tools developed in the UK, the concept is still a key part of the scheduling algorithms used in the Micro Planner range of software[15].

Activity-on-Arrow diagramming became complicated when projects had multiple resource types and multiple identical activities usually differing only in their physical location. To keep the correct logical relationships most of the nodes had to be split by using 'dummy' arrows. In a 'ladder-feed' diagram for a pipeline or roadway segmented into discreet sections, there could be as many 'dummy' arrows as work activity arrows. The use of the logic-splitting 'dummy' arrows had to be precise. Figure 15 is an edited version of this type of schedule and for each double node [OO] there is also a logic-splitter 'dummy' arrow, [O->O] that is not drawn.

---

13 A full discussion of the different constructs that can cause the overall schedule duration to change differently to the change in a task duration are discussed in ***Critical confusion – when activities on the critical path don't compute……***, see: https://mosaicprojects.wordpress.com/2016/06/12/critical-confusion-when-activities-on-the-critical-path-dont-compute/

14 ICL 1500/4 PERT included Ladders on its release in May 1963. The documentation suggests Ladders were part of the 1500/3 PERT program (1962) with only minor improvements in the /4 release.

15 For more on **Micro Planner** see: http://www.microplanning.com.au
The assistance of Micro Planning International's Raf M. Dua in providing information on Ladders is acknowledged.

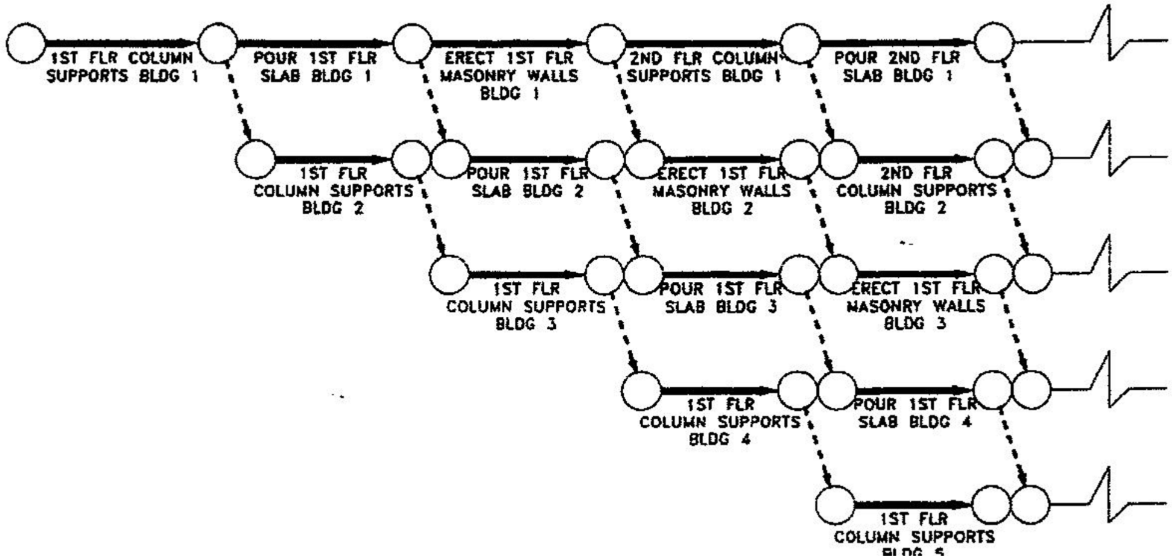


Figure 15 – A typical progressive feed problem

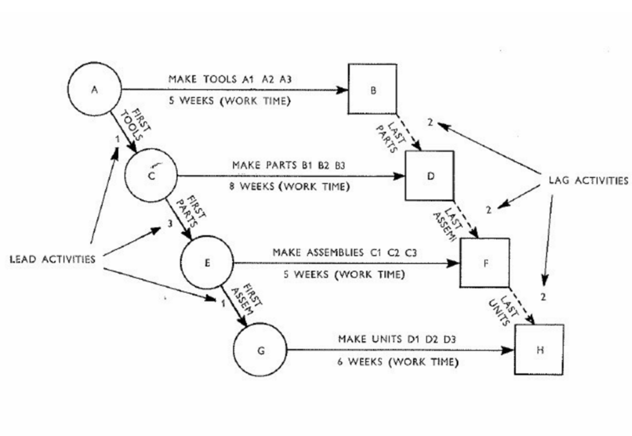
Precedence diagrams are not much better; using normal links, SS only controls the start relationship, FF only controls the finish relationship and whilst combining SS and FF provides the best control, only the ends (or start and finish events) of the tasks are linked and problems similar to the one defined in Fig. 14 above can easily occur.

Ladders are different! The concept of a 'Ladder' moves the management of overlapping activities forward to incorporate the idea of 'progressive feed'.

Ladder activities were developed as a special group of activities that are used to represent progressive feed tasks. An example of a progressive feed task occurs in the manufacture of a number of identical components, each component having to go through several processes such as manufacturing, assembly and testing. To represent these processes in a network in the normal way would require one activity for the manufacture of each component, another to assemble the unit, probably another for inspection, etc. The same sequence of activities would have to be repeated for each unit. The resulting network could be extremely complex; ladders simplify the representation of the work.

Rung activities are the various tasks to be undertaken with defined durations, resource requirements, etc but designated as a 'rung' type of activity. The leads and lags are special activities specified with reference to the rung activity from which they originate. Before the second task in such a progressive feed process can start, the first task must have been in progress for a given time to ensure a supply of components for the second task. The time that must elapse before the second task starts is called lead time. Similarly, there is a lag time after the completion of the first time before the second task can be completed.

A ladder in an ADM network, and a representation of the 3 components in a time scaled network:



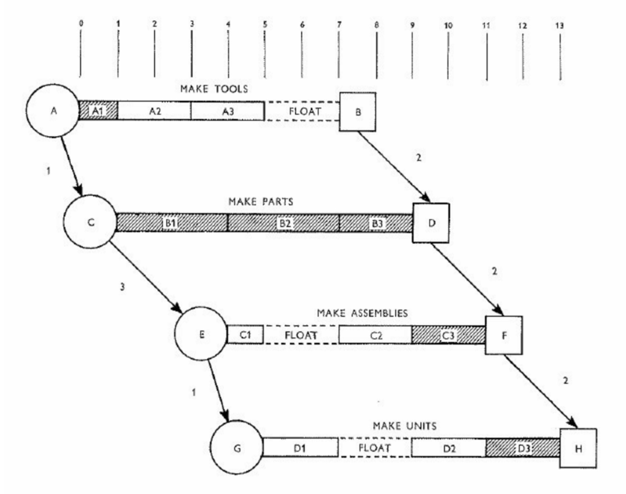


Figure 16 - A Ladder

This is similar to the operation of SS and FF links, however, from an analytical viewpoint, the major advantage of a ladder is if work stops on one rung, the delay is automatically flowed through to the work on all of the dependent rungs, not just the end event.

Progressive Feed Links

The Metra Potential Method (MPM) introduced a number of additional link types that can now be seen in some advanced PDM network tools. These links use the concept of progressive feed in the same way the ADM 'ladder' described above. Depending on the tool, the degree of overlap between two activities can be managed based on either a percentage complete or a set duration. In both cases, the leading activity must stay the designated amount in front of the succeeding activity and if the lead activity stops (eg, as a consequence of resource analysis), the succeeding activity stops as well.

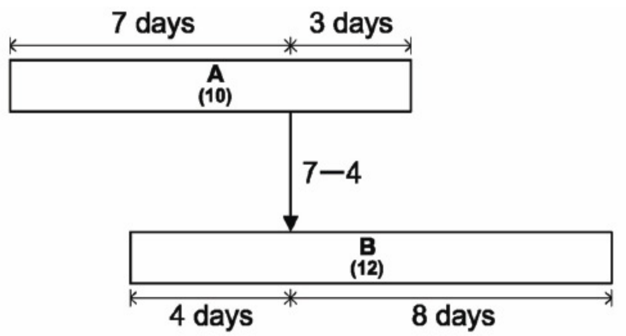
- ACOS+1 uses the AP link type, AP=3 means the succeeding task cannot start until 3 days after the start, and cannot finish until 3 days after the completion of the predecessor.
- Deltek Open Plan allows percentage lags on all link types. The leading task needs to maintain the specified percentage completion ahead of the successor. A 20% lag means that if the predecessor is 60% complete, the maximum completion on the successor is 40% (it may be less but cannot be greater).
- Projack has a 'continuous relationship' that maintains a consistent overlap between predecessor and successor.
- Spider Project allows the concept of a 'Volume Lag', in pipeline construction trench excavation shall be done before lowering pipes but these activities can be done in parallel as long as the trenching crew and the lowering crews work at certain distance from one another. This is typical laddering relationship - both a minimum lag and a maximum lag can be defined. This relationship is physical: the distance between crews shall be no less than 100 meters (for safety) and no greater than 500 meters (to prevent too much trench being opened). This type of relationship is called a 'double link' in Spider.

The precise way these capabilities are incorporated into various tools differs. Planners and schedulers need to be fully aware of precisely how the options function before using them.

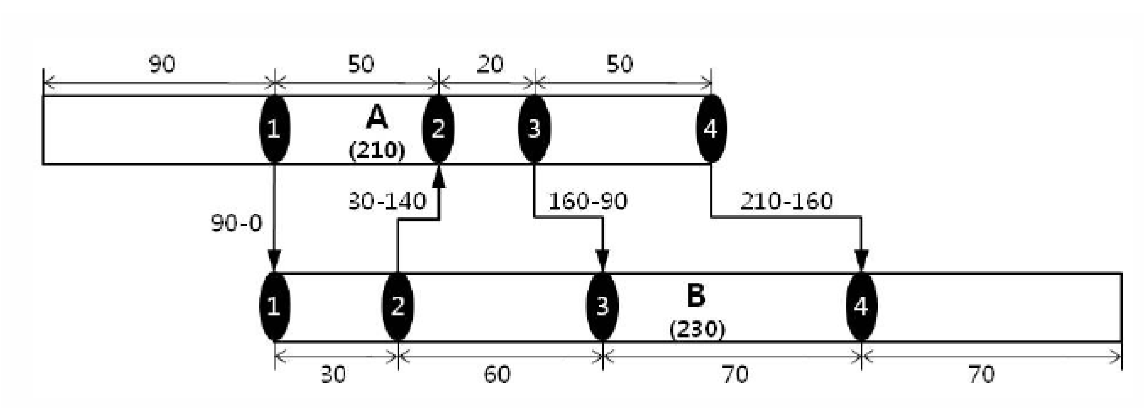
Other Approaches to Managing Overlapping Tasks

Beeline Diagramming Method (BDM)

The concept of Beeline is to represent the overlapping relationship between two consecutive tasks by the shortest straight line (the beeline). BDM connects any point in the predecessor to any point in the successor.



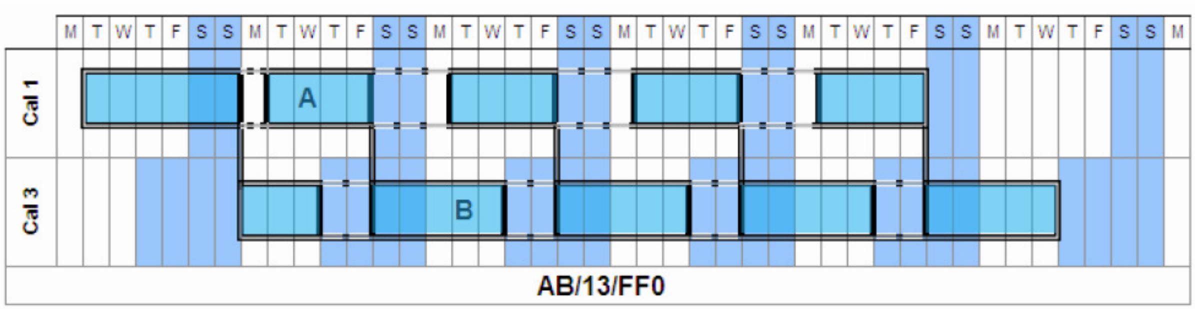
Multiple links are allowed:



For more on BDM see: https://www.mosaicprojects.com.au/PDF-Gen/Beeline_Diagramming.pdf

Chronographic Method

The Chronographic Model uses the concept of internal divisions and internal measurement as a function of production, referred to as the Temporal Function, which has the effect of delaying or anticipating the start of the second activity in order to respect the predecessor production, taking into account the different calendars the various activities may be working to.



For more on the Chronographic Model see:
https://www.mosaicprojects.com.au/PDF-Gen/Chronographic_diagramming_method.pdf

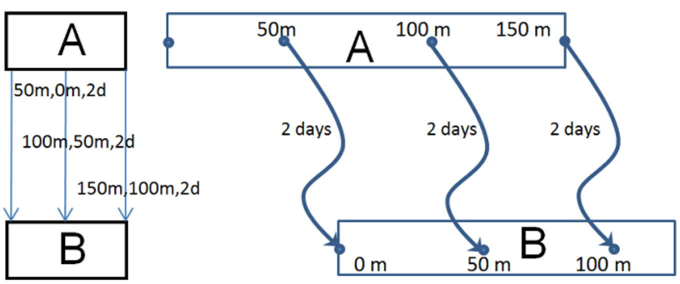
Relationship Driven CPM

RDCPM®, the Relationship Diagramming Method (RDM) variation of the Critical Path Method of schedule analysis focuses on the reason for the relationship between activities and the reason for their overlap. Links can originate at external (end) events or internal events within an activity. A wide range of link types are supported. A similar approach to RD-CPM is embedded in the Graphical Path Method (GDM) where the connected internal points are called embedded nodes[16].

For more on RD-CPM see: https://mosaicprojects.com.au/WhitePapers/WP1035_RD-CPM.pdf

Point-to-point relationships

Point-to-point relationships seek to combine the best elements of the above concepts into a single theory. A point-to-point relationship can connect any two points of related activities with minimal or maximal time lag. Points can be defined using time or volume.

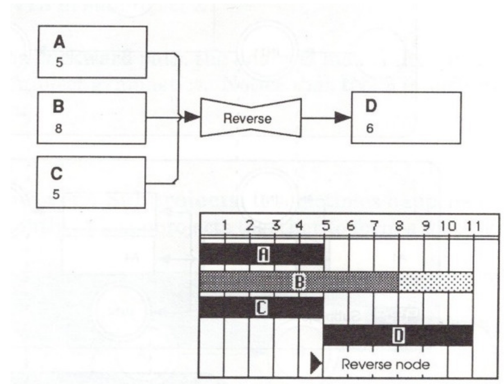


In the above example, (50m,0m,2days) means that 2 days after the completion of the first 50m of the predecessor the successor can start. Standard PDM end to end connections (FS, SS, FF, SF) simply become an allowed subset of this relationship type.

The adoption of any of these 'new' link types into general practice will affect the fundamentals of scheduling; all existing definitions, generalizations, and calculations of floats, the critical path, the classification of critical activities, and the algorithms for resource optimization, etc., will need to be adapted.

---

16 For more in GPM see: http://pmatechnologies.com/tutorials/graphical-path-method/



Reverse Logic Event

Micro Planner X-Pert offers a **Reverse Logic Node** changes the normal network logic that requires all of the preceding tasks linked to the node to be completed and/or the link constraints to be completed, before the event is triggered and the following task(s) can start. The Reverse Logic Node allows its successors to start as soon as one of its predecessors in complete.

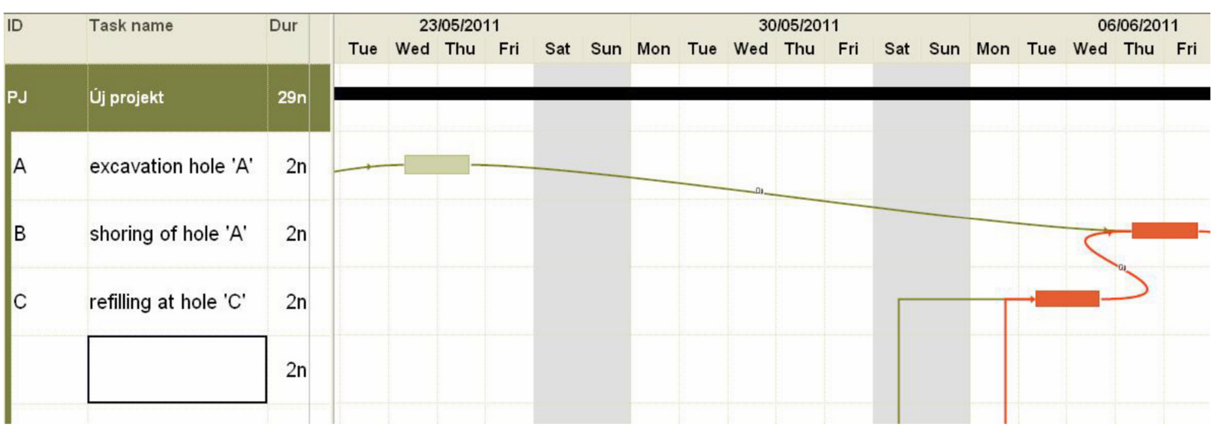
Line of Balance & Chainage Charts

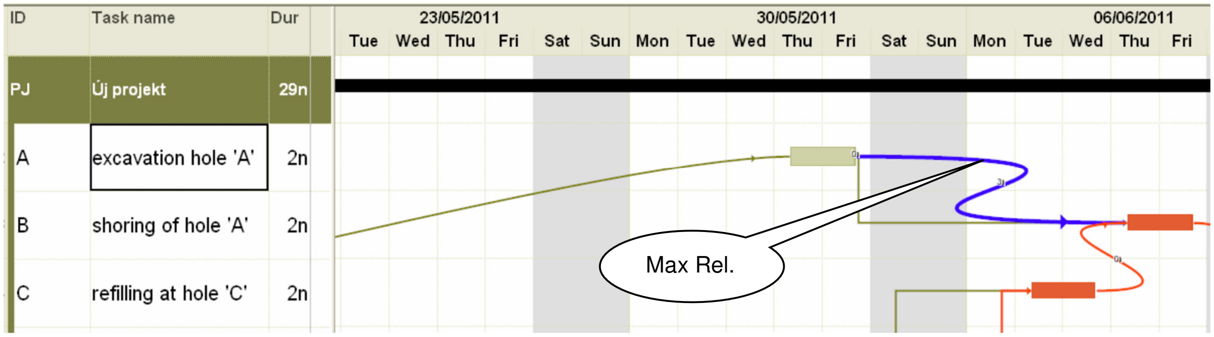
Line of Balance (LOB) is a method of showing the repetitive work that may exist in a project as a single line on a graph. Unlike a Bar Chart, which shows the duration of a particular activity, a LOB Chart shows the rate at which the work that makes up all of the activities has to be undertaken to stay on schedule. This is an alternative approach to network diagramming that works well on linear projects such as pipelines.

For more on LOB see: https://mosaicprojects.com.au/WhitePapers/WP1021_LOB.pdf

Maximum Links

The Metra Potential Method (MPM) also allows the concept of a 'Maximum' relationship. Maximum relationships maxSS, maxFS, max SF, and max FF. Force the following activity to start within a defined period of time after the predecessor. An example would be responding to the people who contributed to a customer survey. The thankyou mail out cannot be sent until after the completion of the survey but should not be delayed too long, by using a maxFS 5day link, the 'thank you' can be sent as soon as the survey is completed or at any time up to 5 days after the survey. But if it has not already started, the 'Send thank you' activity will be forced to start on the 6th day. These links are included in the ACOS[9] system and other European tools based on MPM.

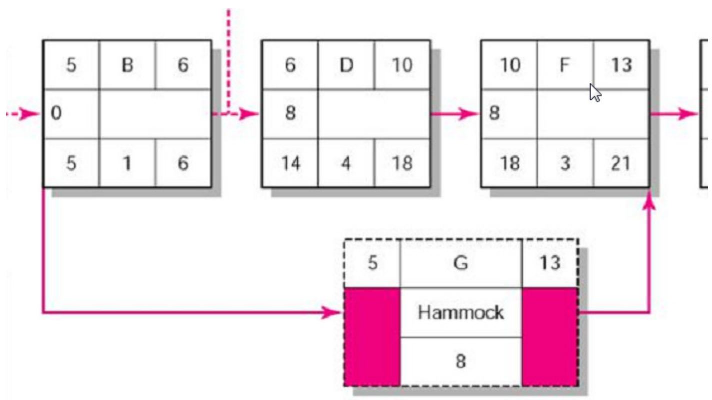




The effect of the blue 'maximum' in the network above is to pull the start of the 'excavation' activity back nearer to the availability of the shoring which is being transferred from 'Hole C'[17]

Hammock Activities

The 'Hammock Activity' is a cross between a link and an activity. The duration of the 'Hammock' is derived from the time between its start connection and its finish connection (it has no predetermined duration) but the hammock can have descriptions, codes and other attributes of a normal activity



Hammocks are very useful for carrying time related costs and determining the duration of supporting activities and equipment needed for a project. However, when using 'Hammocks' it is important to ensure that the Hammock does not become a controlling link in the schedule - the activities 'under' the Hammock should be logically linked from end-to-end.

The example I use when teaching is the time the tower crane is needed on a high-rise construction project. The start of the crane working on-site is driven by the concreting of the foundations and erection of the crane. It is then required through to the time the last heavy lifting to the roof is finished (typically roof mounted plant and equipment) once this activity is finished it can be removed. The duration of the hammock is derived from the timing of these two events and is calculated automatically by scheduling tools that implement hammocks correctly.

[17] Example provided by Hajdu Miklós, Faculty of Civil Engineering and Architecture, Budapest University.

Many software tools that do not have the capability to implement Hammocks and to hide the deficiency confuse a 'hammock' with either a 'Level of Effort' or a 'Summary' task[18]. However, summary tasks are part of the logic structure and summarize lower level tasks within a coding system. Hammocks are not dependent on any coding structure.

The benefit of a 'Hammock' over a Level of Effort (LOE) task is the Hammock's duration is flexible and automatically adjusts to changes to the underlying logic in the schedule, whereas LOE activities have a set duration that requires manual adjustment if the project changes.

Conclusions

The developer of the PDM networking methodology, Dr. John Fondahl, was always of the view the only safe link to use in a precedence schedule was the Finish-to-Start link. Similar warnings are contained in the *PMBOK® Guide* and the PMI *Practice Standard for Scheduling*.

The issues raised in this paper clearly demonstrate the inconsistencies and problems that can develop using S-S and F-F links. However, it is highly unlikely their use will diminish significantly. Therefore, the responsibility must fall to the managers of schedulers, and the schedulers themselves to make sure the logical constructs used in their schedules are both sensible and mathematically correct.

____

[18] For more on ***Hammocks, LOE and summary tasks*** see:
https://mosaicprojects.com.au/Mag_Articles/P016_Hammocks_LOE_and_Summary_Activities.pdf

The papers in this series:

A Guide to Scheduling Good Practice: https://mosaicprojects.com.au/PDF-Gen/Good_Scheduling_Practice.pdf

Attributes of a Scheduler: https://mosaicprojects.com.au/PDF-Gen/Attributes_of_a_Scheduler.pdf

Dynamic Scheduling: https://mosaicprojects.com.au/PDF-Gen/dynamic_scheduling.pdf

Links, Lags & Ladders: https://mosaicprojects.com.au/PDF-Gen/Links_Lags_Ladders.pdf

Schedule Float: https://mosaicprojects.com.au/PDF-Gen/Schedule_Float.pdf

Schedule Levels: https://mosaicprojects.com.au/PDF-Gen/Schedule_Levels.pdf

Schedule Calculations: https://mosaicprojects.com.au/PDF-Gen/Schedule_Calculations.pdf

---

First Published 27th October 2008 – Augmented and Updated