

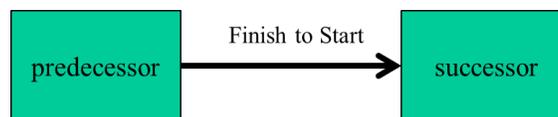
A New View of the Precedence Diagramming Model – Towards a Vectorized Event Scheduling Technique¹

By Crispin 'Kik' Piney

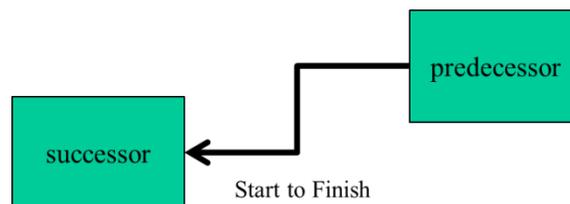
This note does not propose any change to the precedence diagramming method itself. It does however propose a major conceptual change to the way in which the model is viewed.

Background

The standard definition of the PDM is “A technique used for constructing a schedule model in which activities are represented by nodes and are graphically linked by one or more logical relationships to show the sequence in which the activities are to be performed” (Project Management Institute – Practice Standard for Scheduling – Glossary). This is absolutely fine when the logical relationship is “finish-to-start” – i.e. the successor cannot start until a specified time after the predecessor has finished as shown below.



The definition becomes much less applicable, however, for other types of dependency: take for example “start to finish” where the “successor” cannot finish until the “predecessor” has started (see below). In this case, it is confusing, to say the least, to state that the logical relationship shows the sequence in which the activities are to be performed since it linguistically reverses the arrow of time. In fact, it represents a constraint rather than a sequence for the activities.



¹ *Second Editions are previously published papers that have continued relevance in today's project management world, or which were originally published in conference proceedings or in a language other than English. Original publication acknowledged; authors retain copyright. This paper was originally published in the March 2012 edition of PM World Today. It is republished here with the author's permission.*

This was the initial insight that led, through a few steps, to the development of a more consistent and simpler approach to schedule modelling. This paper follows the same set of steps towards the final conclusion.

The next section explains that the problem comes from confusing “activities” with “events”.

New View

So what is the answer? The answer is that it is not the *activity* itself that is the predecessor or successor; it is one of the two possible activity *events* – i.e. Activity Start of Activity Finish. The four possible relationships FS, SS, SF, FF correspond to all of the combinations of these two events.

This view is also totally compatible with the major deliverable of the precedence diagramming method: the calculation of the possible dates at which these events can occur – i.e. the Early and Late values of the Start and Finish dates.

The only other conceptual step is to realize that an activity can also be represented by an arrow, from Activity Start to Activity Finish. The duration value of this arrow is the duration of the activity.

This new approach makes explaining and understanding the critical path algorithm much more straightforward.

In addition, it integrates the concepts of “activity on arrow” and “activity on node” scheduling representations in a natural manner into a consistent and generalized model.

This is explained in more detail in the following sections:

- First, the critical path algorithm will be revisited and explained by focusing simply on the “arrows”.
- This then leads to an analysis in more detail of the role performed by the “arrows”.

Explaining the Critical Path Algorithm

Forward Pass: Early dates

Start with relative date zero.

Wherever the date at the origin of an arrow is known, follow the arrow, adding on the “duration” value of the arrow. This gives the value of the end of the arrow (i.e. the earliest date that this successor event could occur, based on this dependency). If this is the only dependency arrow terminating at this activity, then the calculated value is the one to use for the corresponding “early” event of the destination activity. Calculate the other “early” activity event from the formula:

EF-ES = Duration (ES stands for Early Start, EF, Early Finish)

If more than one dependency arrow terminates on the activity, calculate all of the corresponding ES and EF pairs and retain the one with the largest values.

Repeat the calculations until all of the ES and EF have been evaluated.

The value of the final EF gives the best possible duration for the project. Take this as starting point for calculating the “latest” dates.

Backward Pass: Late dates

Reverse the path and the logic of the forward pass as follows.

Start from the end of the project and assign it the value calculated in the forward pass above.

Wherever the date at the head of an arrow is known, follow the arrow back to its origin, subtracting the “duration” value of the arrow. This gives the value of the origin of the arrow (i.e. the latest date that this predecessor event could occur, based on this dependency). If this is the only dependency arrow originating from this activity, then the calculated value is the one to use for the corresponding “late” events of the destination activity. Calculate the other “late” activity event from the formula:

LF-LS = Duration (LS stands for Late Start, LF, Late Finish)

If more than one dependency arrow originates on the activity, calculate all of the corresponding LS and LF pairs and retain the one with the smallest values.

Repeat the calculations until all of the LS and LF have been evaluated.

Characterizing the Components

The new model has only two types of components:

- the events
- the arrows

These are explained in more detail below.

The Events

Events are linked to activities and, as such, there are two main types, and some additional uses as follows:

- 1) Activity-related event types
 - a) activity start, with two scheduling values
 - i) early start
 - ii) late start
 - b) activity finish, with two scheduling values
 - i) early finish
 - ii) late finish
- 2) Additional uses
 - a) project start
 - b) project finish
 - c) project milestone, with two scheduling values²
 - i) earliest possible milestone date
 - ii) latest possible milestone date

The Arrows

The arrows link events and also separate them by representing the duration to be applied from the start to the end on the arrow. Since the arrows in this model have a direction and a length, the term “arrow” will be replaced in all that follows by the more accurate term “vector”.

In this way, a vector can be used to separate any two events in the schedule and represent different scheduling artefacts, as follows:

1. Task: this vector represents an activity
 - a. the two ends correspond to the start and the end of the activity, respectively;
 - b. the length corresponds to the estimated duration.

² This is often overlooked in explanations of PDM

2. Buffer: this vector represents a contingency buffer in general, or a critical chain buffer – Project buffer or Feeding buffer (a few more details for critical chain are given later)
 - a. the length of the vector defines the contingency time that has been made available for all of the preceding activities in the project or in the chain
3. Dependency
 - a. Mandatory dependency: a physical, unbreakable rule (e.g. undercoat prior to top coat when painting)
 - b. Discretionary dependency: applied at present to “improve” the model (e.g. risk reduction, house rules, etc., such a cheese before the fruit course)
 - c. Resource exclusion (based on priority use of scarce resources)

It is useful to add characteristics to each vector

1. The type of duration. Durations can be:
 - a. Fixed (zero or non-zero. E.g. a “lag”)
 - b. Resource-dependent (i.e. based on specified values of effort and assigned resources)
 - c. Inherited (e.g. in a “hammock” where the duration is evaluated from the start and finish events (which are calculated from the rest of the network)
2. Effort required
 - a. for effort-related activities
3. The assigned resources
 - a. Name
 - b. Amount per name
4. Other, such as
 - a. Tracking rules (e.g. the way of defining and using “percent complete” for Earned Value calculations)
 - b. Splittability (can it be interrupted once it has started or does it have to be an uninterrupted duration?).

The way in which these concepts can be used to reduce the overall model to a simple network of vectors and events is explained below.

An activity-free network

If you take this new model to its logical conclusion, you can effectively remove the activity node from the model by considering just the activity start and activity finish events. These events are linked by a Start-Finish dependency vector specifying the activity duration. This provides a new scheduling method as follows:

Vectorized Event Scheduling Technique (VEST): A technique used for constructing a schedule model, in which activities are represented by their start and finish events. The events are graphically linked by one or more vectors indicating relationships (events, and inter-event constraints), along with their characteristics such as duration, resource impacts, etc., to specify the sequence in which the events – and, consequentially, the activities – are able to occur.

This approach can then be applied not only to the precedence diagramming method as shown in Figure 1 but also to other classical scheduling methods, as explained below.

The durations are as follows:

- Act 1 = 3
- Act 2 = 7
- Act 3 = 6
- Act 4 = 5
- Act 5 = 1
- Act 6 = 3

The lag and lead values are shown directly on the diagram.

The critical path is shown as follows:

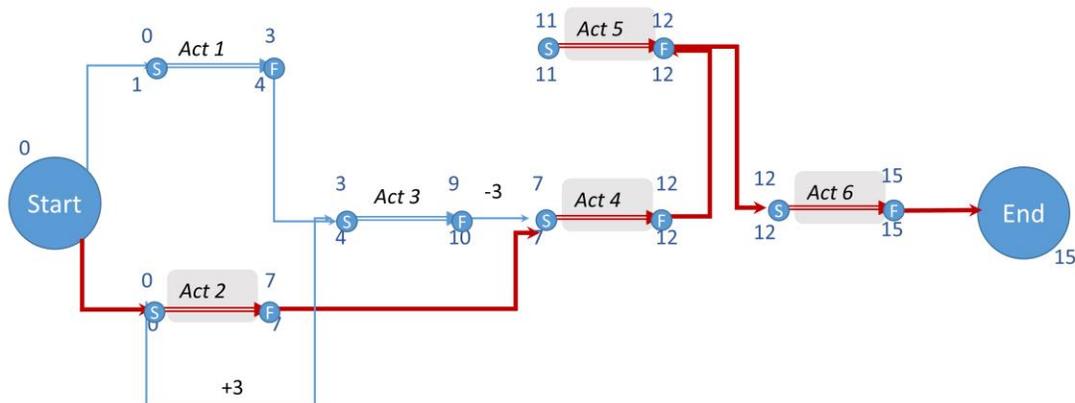


Figure 1: The VEST Applied to a Precedence Diagramming Example

Extensions to other scheduling methods

The following methods can be integrated seamlessly into the VEST:

- The program evaluation and review technique (PERT)
- The Critical Chain
- Milestone charts
- Program planning.

PERT

The approach works directly with the PERT approach once you understand that the PERT uses event-on-node but restricts the type of event to Finish-and-Start, and the only types of vector to:

- a) activity and
- b) dummy (a zero-duration activity vector to allow path splitting with path convergence to be represented in the PERT diagram) – one of the known complicating drawbacks of the PERT model itself, which is removed by using the VEST.

A PERT example is shown in Figure 2.

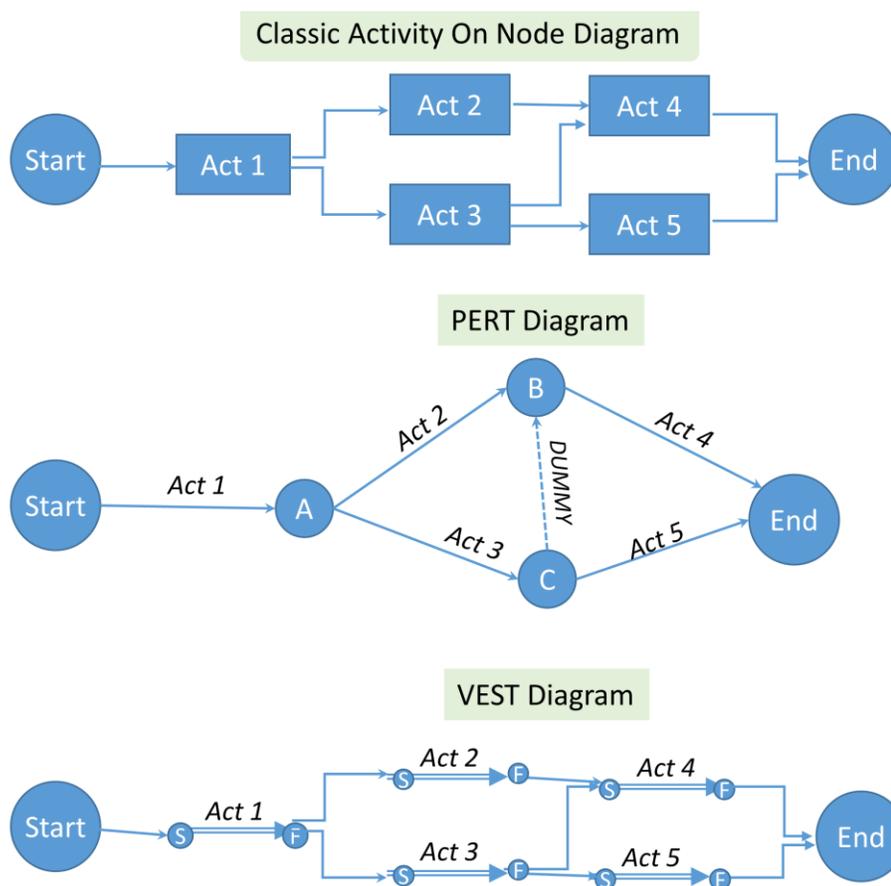


Figure 2: The VEST Applied to a PERT Example

Critical chain

The feeding buffers and program buffer required by the critical chain project management (CCPM) approach can also be incorporated into the VEST without any change to the overall concept: each type of CCPM buffer is represented by an arrow that defines the duration assigned to the buffer, as shown in Figure 3.

The durations
 (more or less halved from the
 “standard CPM”) are as follows:
 Act 1 = 2
 Act 2 = 3
 Act 3 = 3
 Act 4 = 3
 Act 5 = 1
 Act 6 = 2

The CCPM buffers are (about half the corresponding chain):
 Feeding Buffer (FB) = 2
 Program Buffer (PB) = 5

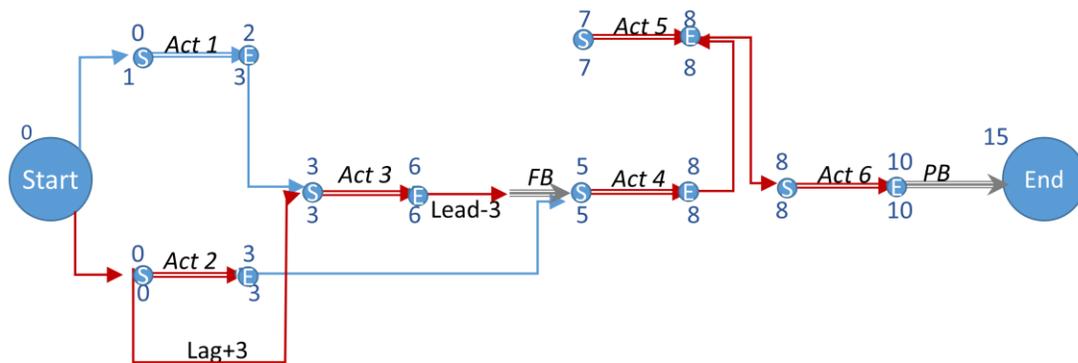


Figure 3: The VEST Applied to a Critical Chain Example

Milestone charts

A milestone chart is in fact a classical event-on-node chart of the PERT variety.

Program scheduling

The VEST diagram has no problem including project and non-project activities together as the basis for creating and calculating the program roadmap.

A detailed example of applying the VEST to Program Management is presented in detail in Kik’s book “Earned Benefit Program Management” that is scheduled to be published by Easter 2017.

Conclusion

Once you understand that the Precedence Diagramming Method terminology applies to activity *events*, rather than just “the activity”, the simplification provided by the proposed Vectorized Event Scheduling Technique becomes obvious, the existence and action of the different types of dependencies (FS, SS, SF, FF) becomes clear, and the algorithm for solving the network becomes easier to understand. The VEST is directly applicable not only to Activity on Node Precedence Diagramming, but also to Activity on Arrow, Event on Node and Critical Chain.

Other definitions

For completeness, all of the elements of this method can be defined and integrated with some from the Precedence Diagramming Method, as follows:

Event: A significant point in the schedule model.

Activity event: A significant point in the execution of an activity (also abbreviated to “event”). There are two sorts: the Start Event and the Finish Event.

Start event: The point at which the execution of an activity begins.

Finish event: The point at which the execution of an activity completes.

Successor Event (replaces the PDM concept of Successor Activity): An event that logically follows a given activity event in the schedule. Note: It is normally used to mean “direct successor” rather than any of the subsequent events (i.e. the successors to successors are not to be considered “successor events” in the context of this definition).

Predecessor Event (replaces Predecessor Activity): An event that logically leads into a given activity event in the schedule. Note: A similar comment to the note on Successor Events applies.

Project Schedule Network Diagram: A graphical representation of the logical relationships between the events.

Relationship vector: An arrow indicating an event-to-event precedence constraint, along with the delay that this relationship imposes – such as: an activity duration, a lag, a contingency buffer, etc.

Event-to-Event: A logical relationship in which one event cannot occur until the predecessor event has occurred. For activities, there are four such relationships: Start-to-Start (SS), Finish-to-Start (FS), Start-to-Finish (SF) and Finish-to-Finish (FF).

Activity Duration: The time separating the start and finish events of an activity.

Late Date: In the Critical Path Method, the latest point in time at which the corresponding (start or finish) event could occur without entailing a modification to the project end-date based on the existing schedule model. For activity scheduling the Late Start and Late Finish dates are calculated.

Early Date: In the Critical Path Method, the earliest point in time at which the corresponding (start or finish) event could logically occur based on the existing schedule model. For activity scheduling the Early Start and Early Finish dates are calculated.

The following definitions do not need to be adapted to the new, event-based view. Although they may be defined correctly elsewhere, they are included here to ensure completeness:

Schedule Model: A representation of the plan for executing the project's activities including durations, dependencies and other planning information, used to produce a project schedule along with other scheduling artefacts.

Project Schedule: A version of the schedule model in which all of the activity information is uniquely specified for execution.

Schedule Baseline: The approved version of the schedule that is used to drive activity execution and project performance tracking; it can only be changed through formal change control procedures.

Note: the following definitions do not need to be changed and are defined in most descriptions of the critical path method:

- Critical Path
- Critical Path Activity
- Critical Path Method
- Forward Pass
- Backward Pass
- Path Convergence

- Path Divergence

Conclusion

The new approach – the Vectorized Event Scheduling Technique – removes the inconsistencies that appear in all of the other scheduling models without invalidating any of these models. It gives a simplified, logically consistent view of project schedule planning that is compatible with all of the existing project management methodologies and as such is a useful conceptual change.

Future prospects

Although it will require further work and analysis, it does look as if the VEST can be expanded beyond the normal deterministic scheduling approach by extending the concepts of vectors and of events to include uncertain conditions and events:

- Addition of a probability value or function to the parameters stored with each vector, as well as allowing control to loop back will provide an effective basis for incorporating uncertainty within the flow of control – and therefore the duration and cost – of a project. There can be two type of probability: a) the likelihood of the existence of the vector and b) the probability distribution function for vector values such as duration or cost. The probability values can either be fixed or a function to be evaluated based on other values within the network – thereby allowing, for example, correlation between events. Simulation tools can be used to evaluate the resulting probability distribution curves.
- Inclusion of “conditional events” (external milestones) as triggers within the network, each with a specific probability of occurring, can be used to add external risks to the network without disruption the overall concepts or algorithms. In the same way as for the vector probabilities, these event probabilities can either be fixed or affected by other values within (or beyond) the schedule model. For example, the probability of occurrence of one event could depend on the number of times some other event occurs.

About the Author



Crispin Piney

Author



After many years managing international IT projects within large corporations, **Crispin (“Kik”) Piney**, B.Sc., PgMP is now a freelance project management consultant based in the South of France. At present, his main areas of focus are risk management, integrated Portfolio, Program and Project management, scope management and organizational maturity, as well as time and cost control. He has developed advanced training courses on these topics, which he delivers in English and in French to international audiences from various industries. In the consultancy area, he has developed and delivered a practical project management maturity analysis and action-planning consultancy package.

Kik has carried out work for PMI on the first Edition of the Organizational Project Management Maturity Model (*OPM3™*) as well as participating actively in fourth edition of the *Guide to the Project Management Body of Knowledge* and was also vice-chairman of the Translation Verification Committee for the Third Edition. He was a significant contributor to the second edition of both PMI's Standard for Program Management as well as the Standard for Portfolio Management. In 2008, he was the first person in France to receive PMI's PgMP® credential; he was also the first recipient in France of the PfMP® credential. He is co-author of PMI's *Practice Standard for Risk Management*. He collaborates with David Hillson (the “Risk Doctor”) by translating his monthly risk briefings into French. He has presented at a number of recent PMI conferences and published formal papers.

Kik Piney can be contacted at kik@project-benefits.com.