

Improving the Teaching of Project Management Software Tools Using Scaffolding Based on Cladistics Parsimony Analysis

Stanislaw Paul Maj¹

¹ Engineering Institute of Technology, Western Australia

Correspondence: Stanislaw Paul Maj, Associate Dean (Research), Engineering Institute of Technology, 1031 Wellington St., West Perth, Western Australia, 6005. Tel: 1300-138-522. E-mail: paulm@eit.edu.au

Received: February 14, 2017

Accepted: March 9, 2017

Online Published: April 8, 2017

doi:10.5539/mas.v11n5p73

URL: <https://doi.org/10.5539/mas.v11n5p73>

Abstract

Project management software tools are essential tools for the different phases of projects. However a number of software tools exist ranging from the relatively simple to those that extremely feature rich. Feature rich software tools have their advantages due to their wide repertoire of options but can be difficult to learn. Ideally the main steps in project management need to be identified and mapped. The five layer domain analysis model was used (but only the first two layers) in conjunction with cladistics and parsimony analysis to produce a diagrammatic, modular abstraction based on hierarchical top down decomposition. This model contextualizes the steps in project management by means of a high level overview that can be decomposed into greater detail. Significantly it is universally applicable to software management tools.

Keywords: project management software, pedagogy, scaffolding

1. Introduction

Project management is supported by the availability of software tools – both free and at a cost. These software tools allow a project to be planned and then tracked. As such in the planning phase they allow the entry of basic parameters such as: start and end dates; calendars (standard, night and 24 hour) and associated supporting information such as name of project manager etc. High level tasks (activities) can be defined along with the associated parameters such as: duration; milestones; dependencies (finish to start, start to start, finish to finish, start to finish); scheduling (flexible, semi-flexible and inflexible). Each high level task can then be decomposed into subtasks when appropriate. Identified resources (work, materials and cost) can be entered and then linked to the defined tasks. Advanced planning techniques consist of identifying how fixed duration work may be reduced by duration or units and hence align with resource capacity allocation (under, fully or over allocation). Prior to project initiation and hence tracking it is essential to baseline in order to track parameter such as variance, actual and remaining. With project parameters entered some software tools, such as Microsoft Project, not only provide a wide range of views (Gantt, network, critical path etc.), but also numerous other tools such as financial tracking methods (planned value, earned value, actual cost, cost performance index etc.). There are two problems. Firstly knowledge of project management ideally should be generic and not product specific. The Graphical User Interface (GUI) of Microsoft Project and ProjectLibre differ (Figure 1, figure 2) have some similarities. Common tabs between these two products are: file, task, resource and view. However selection of the same tab for each product reveals different menu options. Secondly for feature rich software tools such as Microsoft Project the complexity of these tools must be controlled in order to facilitate learning by novices. In the case of Microsoft Project when some tabs are selected the user is presented with, in some cases, an extensive number of options (figure 3).

2. Method

Complexity can be controlled by the use of abstraction The ACM/IEEE Computing Curriculum 2001 listed abstraction as one of the twelve recurring and fundamental concepts (Tucker et al., 1991). Model characteristics include: diagrammatic, easy to use and employ top down decomposition. Top down decomposition is the property of being able to progressively break a system down into subsystems whilst still maintaining contextual links. In effect abstraction based models consist of both high level and low level integrated representations. According to Cooling, ‘*Good high-level diagrams are simple and clear, bringing out the essential major features of a system*’ (Cooling, 1991). In effect high-level diagrams help define what the system does and hence assist in

navigation. Low-level diagrams handle details. By example, a Personal Computer (PC) is can be described using different models such as: semiconductors; transistors, digital circuits etc. Each model progressively hides technical detail and complexity. Rapid advances in PC technology resulted in the need to define a new high level abstraction that was used as the basis of teaching PC architecture (S. P Maj, Veal, & Duley, 2001).

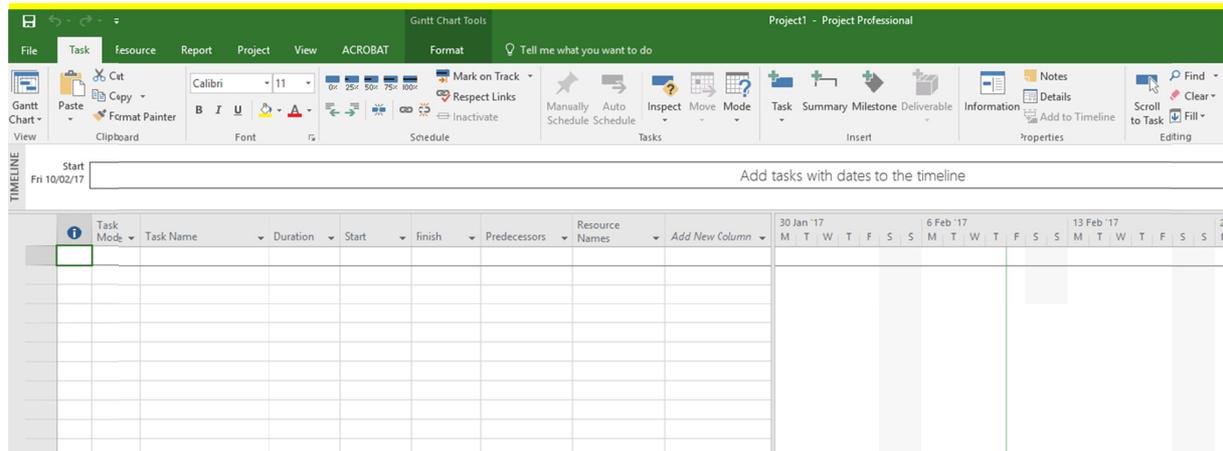


Figure 1. Microsoft Project GUI

In the case of software functionality content structures are an appropriate method of modeling these systems because they convey structural knowledge by means of hierarchical structured relationships (Meyer, 1985). However other modelling methods do exist such as ‘frames’ and ‘slots’(Minsky, 1975).

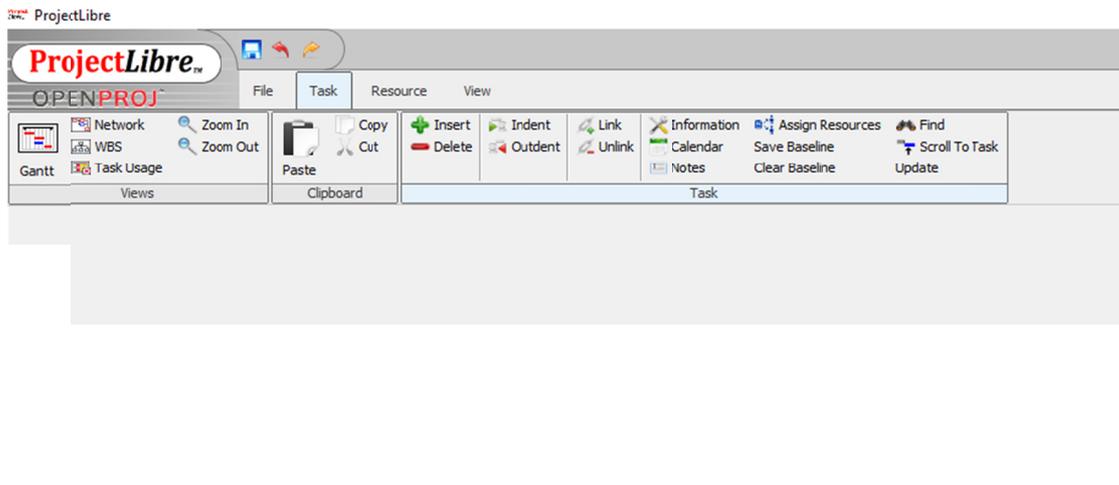


Figure 2. ProjectLibre GUI

For complex systems a major problem is the number of interacting subsystems. Even with a relatively small number of subsystems there are a significant number of combinatorial representations. In order to address this problem Maj proposed the use of cladistics and parsimony analysis (S. P. Maj, Veal, D., 2010). This method defines a structured relationship with a minimum cognitive load. Significantly it is a quantitative method and hence different structured relationships may be constructed and quantitatively analyzed in order to determine the optimal structured relationship model. When instructional material is based on this method there is evidence to show that there are significant improvements in teaching and learning outcomes (S.P. Maj & Veal, 2007). Furthermore this cognitive science approach may be aligned with the SOLO learning taxonomy (S. P Maj, 2017).

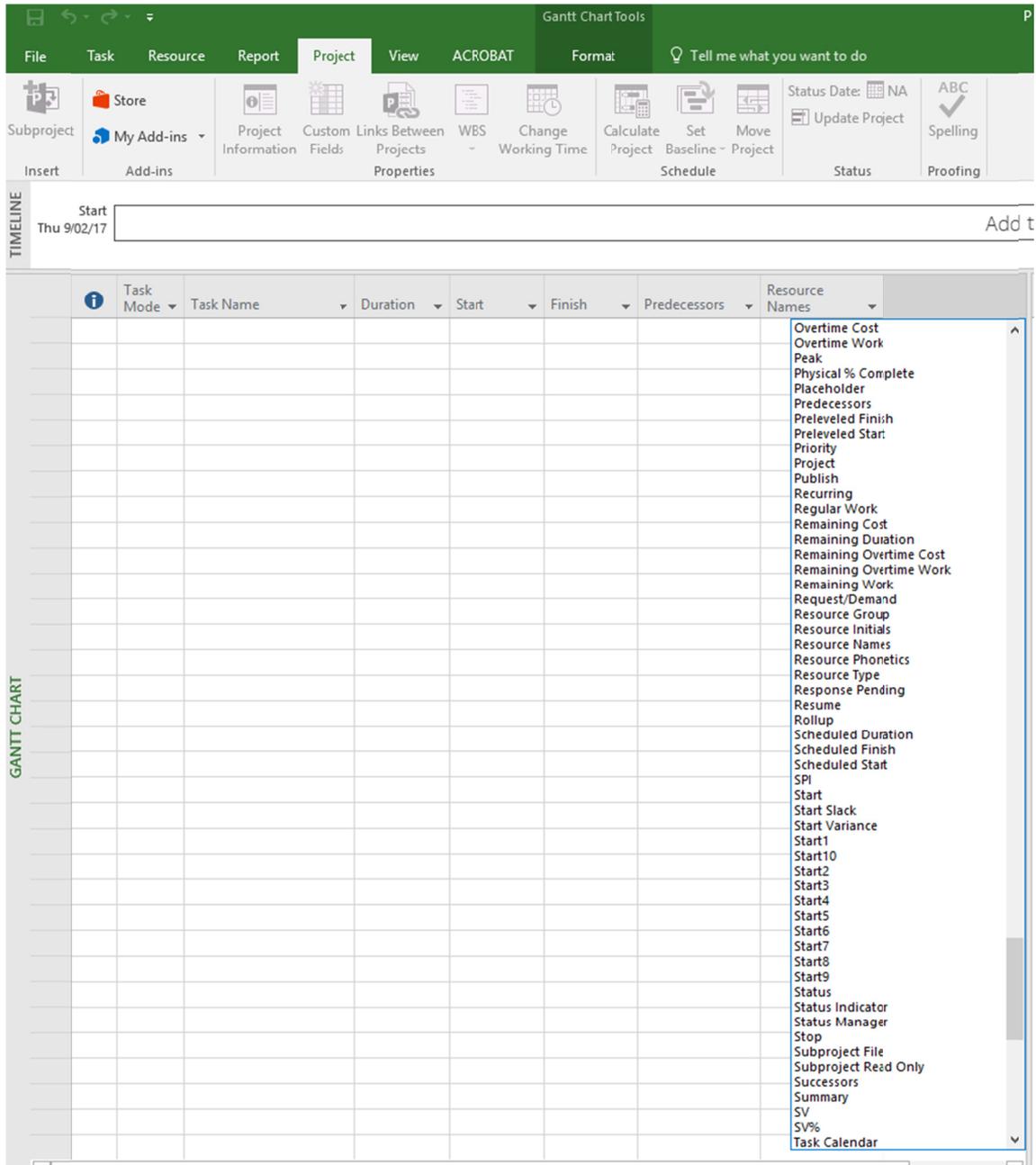


Figure 3. Microsoft Project Pulldown options

Evidence Centered Design provides a five layered model for domain analysis, modelling and assessment (Mislevy, 2003). The domain analysis and modelling layers were the basis of the conceptual modelling. This was complemented by the use of parsimony analysis in order to define the optimum learning sequence (S. P. Maj, Veal, D., 2010), (S. P Maj, 2017). The highest level diagram defines the two phases of project management – planning and tracking (figure 4).



Figure 4. Level 1 - highest

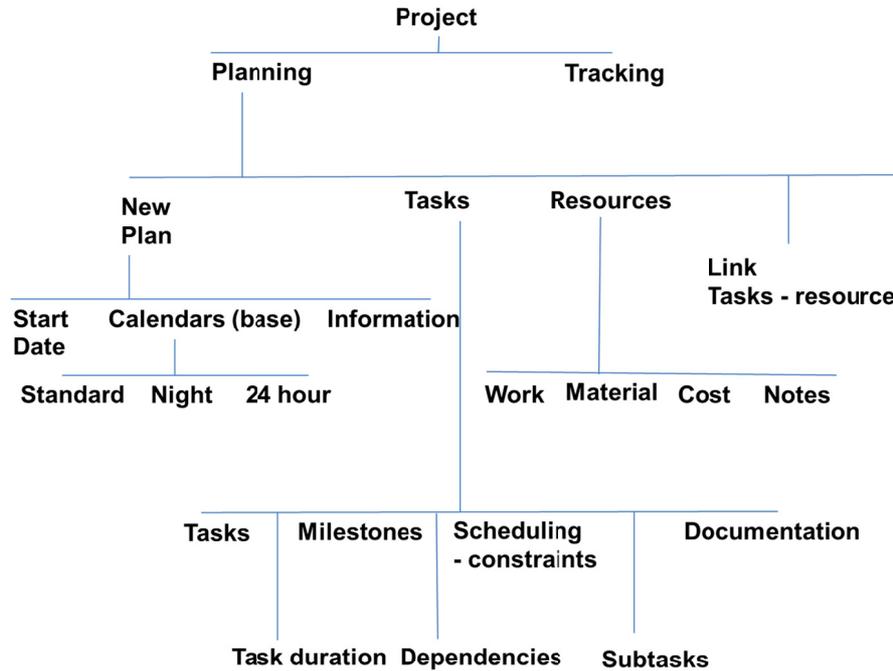


Figure 5. Level 2

This high level diagram is used as the template to provide further detail for example in the planning phase (figure 5). This level 2 diagram provides a list of entries that should be made but contextualizes them in the planning phase.

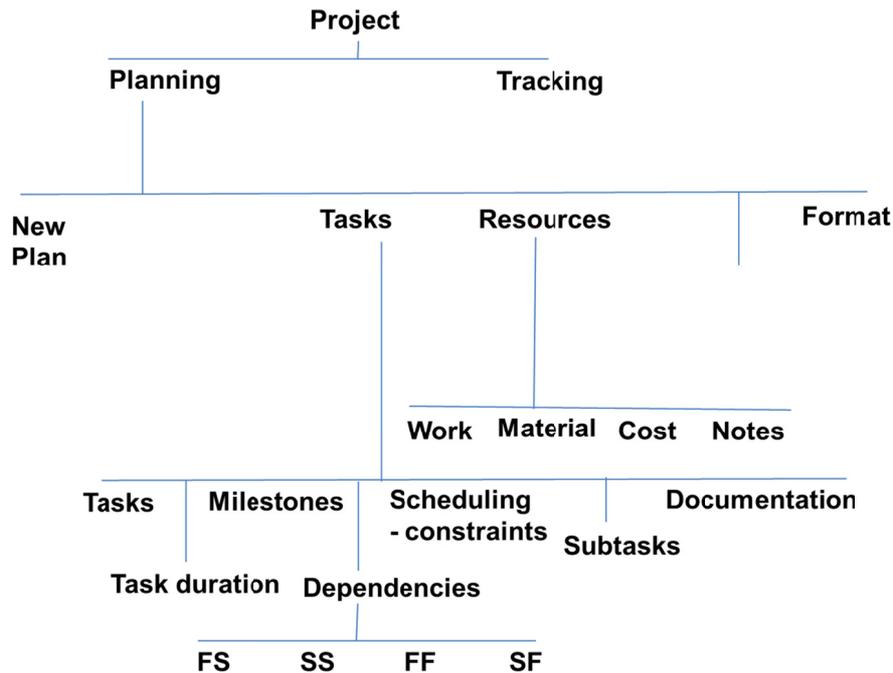


Figure 5. Level 3

It is possible to decompose some of these entries. For example there are four types of dependencies: finish to start (FS), start to start (SS), finish to finish (FF) and start to finish (SF). Similarly there are three types of scheduling constraints (flexible, semi-flexible and inflexible) each with their associated options: start/finish As

Late as Possible (ALP); start/finish As Soon As Possible; Start No Earlier Than (SNET); Start No Later Than (SNLT); Finish No Earlier Than (FNET); Finish No Later Than (FNLT); Must Start On (MS) and Must Finish On (MF) (figure 6). Aspects of this hierarchical mapping can be applied to subtasks. For example, a subtask (at a high level) may consist of multiple sub-subtasks. For each high level sub-task the mapping can be used to identify: duration, milestones, dependencies, and scheduling constraints. Similarly it is possible to map all the tabs and associated pull-down options by means of a hierarchical, top down mapping. Microsoft Project is feature rich and hence offers many configuration options. Hence the mapping for Microsoft Project employs 'zoom in' and 'zoom out' which provides an overview of all functions but allowing increasing levels of detail that importantly still maintain context. Being a diagrammatic model it has the characteristic of concurrency in which relationships between function can all be explicitly identified on a single diagram. This concurrency holds true for all levels of the diagrams.

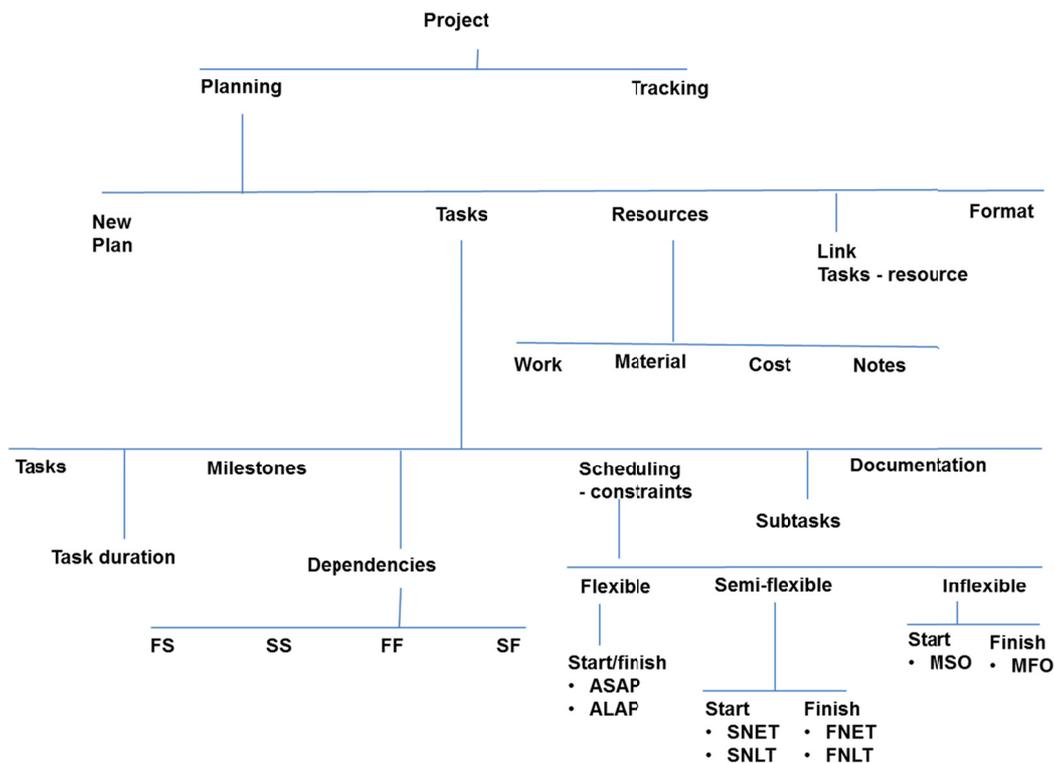


Figure 6. Level 3

3. Results

This hierarchical abstraction model has the following characteristics:

- Diagrammatic
- Easy to use
- Modular
- Uses hierarchical top down decomposition
- Provide an overview and increasing levels of detail whilst maintaining context
- Explicit relationships
- Universally applicable to different project management software tools.

Significantly it can be used as the basis of teaching different project software management tools such as Microsoft Project and Project Libre. Each identified element represents a teaching and learning outcome with the associated power point slides and software exercise.

4. Discussion

Preliminary results strongly support the use of this model however it has only been evaluated for Microsoft Project and ProjectLibre.

References

- Cooling, J. E. (1991). *Software Design for Real-Time Systems*. Padstow, Cornwall: Chapman and Hall.
- Maj, S. P. (2017). Improving Teaching and Learning Outcomes - A novel cognitive science approach. *Modern Applied Science*, 11(1), 264-269.
- Maj, S. P., & Veal, D. (2007). State Model Diagrams as a Pedagogical Tool - An International Evaluation. *IEEE Transactions on Education*, 50(3), 204-207.
- Maj, S. P., Veal, D., & Duley, R. (2001). *A Proposed New High Level Abstraction for Computer Technology*. Paper presented at the 32nd Technical Symposium on Computer Science Education, Charlotte, North Carolina.
- Maj, S. P., & Veal, D. (2010). Parsimony Analysis - a novel method for structured knowledge representation for teaching computer science. *Modern Applied Science*, 4(11), 3-12.
- Meyer, B. J. F. (1985). Signalling the structure of text. In D. H. Jonassen (Ed.), *Technology of Text* (Vol. 2). Englewood Cliffs: NJ: Educational Technology Publications.
- Minsky, M. (1975). A framework for representing knowledge. In P. H. Winston (Ed.), *The psychology of computer vision*. New York: McGraw-Hill.
- Mislevy, R. J., Almond, R. G., Lukas, J. F. (2003). *A Brief Introduction to Evidence-centred Design*. Retrieved from NJ:
- Tucker, A. B., Barnes, B. H., Aiken, R. M., Barker, K., Bruce, K. B., Cain, J. T., . . . Mulder, M. C. (1991). A Summary of the ACM/IEEE-CS Joint Curriculum Task Force Report, Computing Curricula 1991. *Communications of the ACM*, 34(6).

Copyrights

Copyright for this article is retained by the author(s), with first publication rights granted to the journal.

This is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>).