

# Automatic Layout of Project Plans Using a Metro Map Metaphor

Jonathan M. Stott  
University of Kent  
Canterbury, England  
[jms30@kent.ac.uk](mailto:jms30@kent.ac.uk)

Peter Rodgers  
University of Kent  
Canterbury, England  
[P.J.Rodgers@kent.ac.uk](mailto:P.J.Rodgers@kent.ac.uk)

Remo Aslak Burkhard  
University of St. Gallen  
Switzerland  
[Remo.Burkhard@unisg.ch](mailto:Remo.Burkhard@unisg.ch)

Michael Meier  
vasp Datatecture GmbH  
Switzerland  
[mm@vasp.ch](mailto:mm@vasp.ch)

Matthias Thomas Jelle Smis  
University of St. Gallen  
Switzerland  
[msmis@web.de](mailto:msmis@web.de)

## Abstract

*In this paper we describe a tool to improve inter-functional communication of project plans by displaying them as a metro map. Our tool automatically lays out plans using a multicriteria system adapted for the application area. Previous studies have shown that displaying project plans using a metro map metaphor as a complementary visualization to Gantt charts attracts and engages individuals, presents an overview of detail and initiates discussions. Creating such a map manually is time consuming. Hence, we have developed a software tool that converts a standard planning format into a metro map visualization and assists designers in generating comprehensible layouts.*

*Our findings may be important for researchers in the domain of human computer interaction, project managers, knowledge visualization and communication scientists.*

**Keywords---** Project planning, metro map layout problem, multicriteria optimization, knowledge visualization

## 1 Introduction

The metro map is a powerful metaphor to use when visualizing data. Metro maps of one form or another are used in towns and cities across the world [9] to visualize the interconnections of railroad and road networks. It is such a prevalent form that people are able to start using the map quickly – be it to plan their route between two stations on the network or find out where else they are able to visit.

The use of a tube map metaphor has been proposed as being intuitive and engaging for other concepts, such as showing the structure of a thesis or academic course [8] or for showing tours on the world wide web [10]. For these applications the benefits have not been measured

with empirical evaluations. We believe it may also be an effective metaphor for visualizing project plans.

At present, Gantt charts are predominantly used for the mapping of projects in organizations (see Figure 1). While they are effective for planning a project they are not effective for communication purposes, especially when different groups are involved (interfunctional communication). The challenges are: how to catch the attention of users, how to provide orientation and a shared vision, how to present an overview and details, and how to initiate discussions and motivate individuals to act. Burkhard and Meier [3] introduced the Tube Map Visualization for projects in an organization and evaluated its strengths and limitations in an empirical study.

Drawing appealing metro maps is a very subjective problem. Different people have different preferences as to their preferred style and even changing seemingly simple things such as the size of labels and thickness of the edges can have wildly varying consequences for the layout of the final map. In the case of creating project plans, it is extremely desirable to find a way to automatically draw the maps – it can take several weeks to create a metro map from scratch by hand; possibly even significantly longer for people without a background in technical drawing. Various attempts have been made to automatically draw metro maps [6][11] and other more general schematic diagrams [4][5]. These approaches either use graph drawing [2] or a more geometric method [1].

This paper opens with a general description of the multicriteria optimization method and describes the modifications required to allow the method to lay out project plans using the metro map metaphor (project maps). A description of our prototype tool is then given followed some initial results. In the conclusion we assess our work to date as well as outlining potential future research.

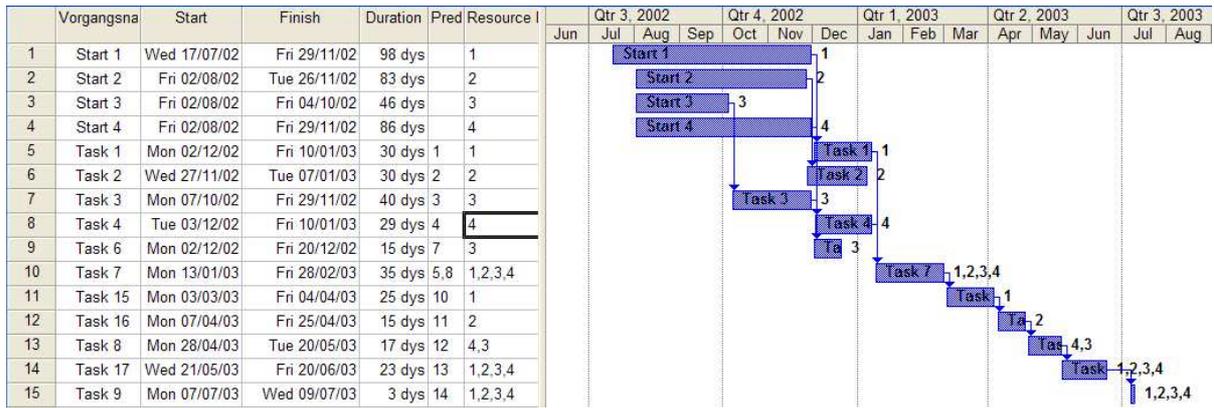


Figure 1. A typical Gantt chart.

## 2 Multicriteria Optimization

The approach to drawing project maps automatically is based on the multicriteria optimization approach taken in [11]. However, the method has to be improved and tuned for drawing project maps. A graph model is used where point features (stations or tasks) are represented as nodes and connections (task dependencies) between nodes are represented as edges.

Our intention is to create a project plan using the metro map metaphor as a complementary visualization to the existing Gantt chart. Each 'line' in the project map corresponds to a particular member or group of members of the project team. As with a Gantt chart, the project map has to have some notion of time. To achieve this, we introduce a horizontal time axis which shows progression of time from left to right. This allows easy comparison of the relative start and end times of various tasks. It also allows us to represent the available time for each task by the difference in  $x$ -coordinates between two task nodes. Dependency of one task on another is implicitly shown by a connection between the two tasks; tasks that are independent of each other will not be directly connected. The graph is embedded on a regular, rectangular grid.

## 3 Basic Metro Map Layout Method

The multicriteria optimization method is based on an iterative hill climber. At each iteration of the method an attempt is made to move each node in the graph within some constrained bounds. Movement is determined based on the calculation of the sum of a set of weighted geometric metrics. A total of five metrics are used:

**Edge length.** This metric is used to penalize long edges (longer than one grid spacing) or very short edges (shorter than one grid spacing).

**Edge orthogonality and diagonality (4-gonality).** We intend to draw edges either orthogonally (horizontally or vertically) or at  $45^\circ$  or  $135^\circ$  diagonal. This metric penalizes edges which are not orthogonal or diagonal.

**Line straightness.** In the case where a line passes through a node (as in a particular metro line passing through a station), the edges of the line either side of a node should be opposite each other. The line straightness

metric penalizes lines which double back more than lines which only make a small bend.

**Edge crossings.** Crossings are assumed to be unwanted, so this metric penalizes each edge crossing.

**Angular resolution.** Where a number of edges are incident to a particular node, the angle between each pair of neighboring edges should be roughly equal. In determining where to move a node, the sum of the weighted metrics is calculated for each potential location that the node can move to. The node will be moved to the location that improves (reduces) the metrics most significantly. If there is no better location for the node, no movement is made.

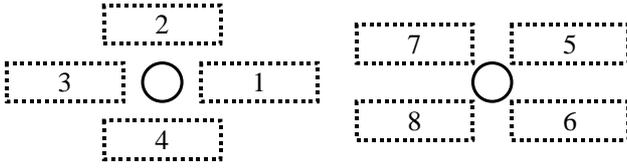
One aim of drawing metro maps is to make the spacing between nodes as even as possible. To necessitate this, two extra techniques were required: a preprocessing step involving the contraction of over-length edges and a way to try to reduce the length of edges that could not otherwise have their length reduced using the hill climber. However, when using multicriteria optimization to draw project maps, long edges convey meaning. In a project map, a long edge will represent a task that has a large amount of time allocated to it and reducing the length of the edge may give a false impression that there is less time to complete the task. We therefore do not use either the preprocessing step or the step involving the reduction in length of over-length edges. Potential conflicts with the five existing metrics also need to be resolved (as mentioned in the Timescale Metric section below).

## 4 Modifications for Project Planning Visualization

Modifications are needed for the project planning application. In particular, we introduce a new metric to take the time axis into account, change the method for initial positioning of nodes, improve the technique for labeling the map and add a way to ensure that the cyclic ordering of edges around a node is preserved.

### 4.1 Timescale Metric

With the introduction of the time axis, it becomes important that task nodes appear correctly in relation to each other. For example, a task that starts before another



**Figure 2. Labelling space for labelling task nodes. Numbers represent the preferential order of labels with 1 being the most preferred location.**

should appear to the left of the other task. For a task node  $n$  with  $x$ -coordinate  $x_n$  and start time  $t_n$  the timescale metric is found by calculating  $(|t_n - x_n|)^2$ . This has the effect of severely penalizing nodes that have strayed a long way from their start time while having little effect for nodes that are close to their start time. As with other metrics, the timescale metric is sufficiently weighted so as to be effective when used in combination with the other metrics.

A side effect of the timescale metric is that the edge length metric is no longer required, as otherwise the two metrics would conflict with each other.

## 4.2 Initial Positioning

When drawing metro maps, the starting position for the multicriteria optimization method was the geographic position of the stations. However this does not apply in the case of project plan data where the diagram is abstract. Initially, all the nodes are placed along a line such that they all have the same  $y$ -coordinate. The initial  $x$ -coordinate of the node is determined by the start time of the task.

## 4.3 Labeling

Labeling is an important aspect of visualizing project maps. Each task node has a label indicating the details for that task. There are also labels indicating the details of each of the lines. We pay attention to the labeling of task nodes.

Our approach to labeling is to integrate it as much as possible into the existing multicriteria optimization method. We developed three labeling metrics which measure the number of intersections between labels and nodes, labels and edges, and labels and other labels. The labeling step was performed at each iteration of the hill climber, after all the nodes had been moved.

We use a similar labeling space as described in [6], with eight potential label positions, ranked in order of preference, to choose from (see Figure 2). For each of the label positions we calculate the weighted sum of the three labeling metrics; the location that we choose will be the one with the lowest weighted sum. If more than one potential positions have equal total weighted metrics, we use the most preferable one according to the ranks of the locations.

## 4.4 Preservation of Cyclic Ordering of Edges Around a Node

In some cases, it is possible for the meaning of a project map to change significantly if the position of two incident nodes (or cluster of nodes) around another node changes. To prevent this from happening, we have introduced a rule to ensure that the cyclic ordering of edges incident to a node remains unchanged when a node is moved.

## 5 Prototype Tool

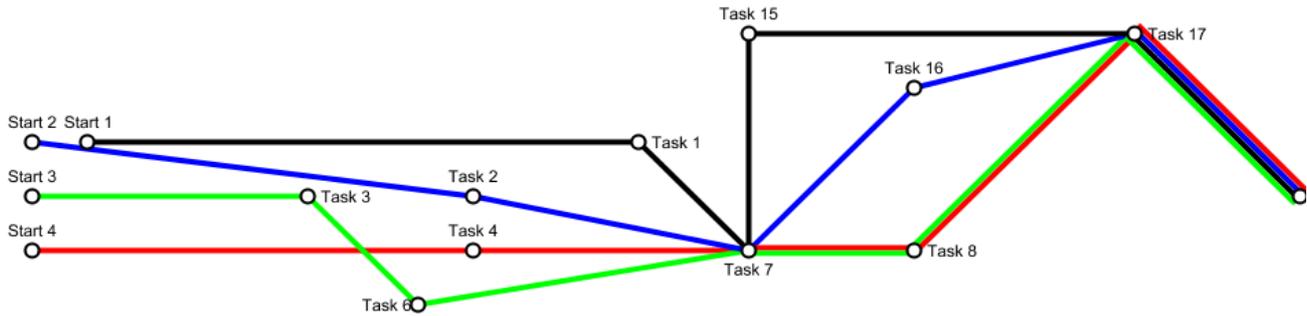
To demonstrate how our method of multicriteria optimization works for drawing metro map style project plans, we have implemented our method in a tool that interfaces with Microsoft Project. It was designed to be as seamless an integration as possible and we intended that no user intervention should be required in order to create the project maps. As such, we needed to find a set of generic metric weightings and choose which features of the method we needed. This is particularly important, seeing as it is highly undesirable that users require any knowledge of how the project maps are laid out.

The tool opens in a new window and allows the user to modify the layout as required (by moving task nodes). The tool also allows the project map to be saved as an encapsulated postscript file (EPS) which can be printed out and displayed.

## 6 Initial Results

Initial results look promising. Figure 3 shows an example of a project map automatically laid out using a metro map metaphor. The project plan that the map represents is the one shown in Figure 1. Each line is drawn with a different color to allow them to be easily distinguished. Therefore, each member of the project can trace his or her 'route' through the project, and in particular, which order their tasks take place and how they relate to other tasks. It is also clear as to which tasks are important for more than one member of the project (such as Task 7) where the lines from more than one member intersect. Project members working together are also clearly shown where two lines run in parallel (such as either side of Task 8 and between Task 17 and Task 9).

While the automatically generated project map has obvious advantages, it suffers from a number of disadvantages such as unnecessary edge crossings, edges drawn very close to nodes (as with the Start 1 node) and edges which are not orthogonal or at  $45^\circ$  diagonal. These problems could obviously be solved with manual editing of the project map, but this is not really desirable as it is meant to automatically generated usable project maps.



**Figure 3. The resulting project map, automatically laid out using a metro map metaphor. The chart is aligned from left to right along a time axis. Each line represents a group; each station an individual or collective milestone in the project.**

Cosmetic problems are also evident. These include the way in which different lines swap places with each other as they pass through a node (e.g. the lines passing through the Task 8 node) and the way that many parallel lines (as between the Task 17 and Task 9 nodes) are wider than the nodes.

Perhaps the most restrictive problem is the time that is required to draw the project map. Simple maps (shown in Figure 3) can be drawn relatively quickly (in the order of seconds), while larger plans will require significantly more time to be drawn (minutes or even hours). If a real-time output is required this is obviously unacceptable. However, the time needed to draw the project maps is less than it would take for someone to manually draw them.

## 7 Conclusions

Our results show that it is possible to automatically generate metro maps which is a valuable aid for the creation of project tube maps. Automatically drawing graphs in general is a difficult and challenging problem, particularly when trying to find a method that is flexible enough to draw all possible graphs. It is not difficult to see that an example graph can be found that cannot be drawn satisfactorily using a particular method. In the case of the multicriteria optimization approach used in this paper, the lack of flexibility is given about by the need to set the metric weightings as constants. Subtle changes to the metric weightings can vary the resulting graph quite significantly, but a tradeoff was needed in order to avoid user intervention.

In consideration for future work, obviously a number of cosmetic and performance improvements can be made (as detailed in the previous section). With the introduction of the timescale metric, the edge length metric was disregarded; it might be beneficial to retain a modified version of the edge length metric that only takes into account the vertical distance of edges. It would also be particularly useful to perform some kind of systematic analysis into the quality of the maps, as well as evaluating out their genuine value in project planning.

## 8 References

- [1] M. Agrawala, C. Stolte. Rendering Effective Route Maps: Improving Usability Through Generalization. *Proc. SIGGRAPH 2001*. pp. 241-249. 2001.
- [2] G. di Battista, P. Eades, R. Tamassia, I. G. Tollis. *Graph Drawing: Algorithms for the Visualisation of Graphs*. Prentice Hall. 1999.
- [3] R. Burkhard, M. Meier. Tube Map: Evaluation of a Visual Metaphor for Interfunctional Communication of Complex Projects. *Proc. I-KNOW '04*. pp. 449-456. 2004.
- [4] S. Cabello, M. de Berg, S. van Dijk, M. van Kreveld, T. Strijk. Schematization of Road Networks. *Proc. 8<sup>th</sup> Annual ACM Symp. on Comp. Geom.* pp. 33-39. 2001.
- [5] S. Cabello, M. van Kreveld. Schematic Networks: an Algorithm and its Implementation. *Advances in Spatial Data Handling*. pp. 475-486. Springer. 2002.
- [6] J. Christensen, J. Marks, S. Shieber. An Empirical Study of Algorithms for Point-Feature Label Placement. *ACM Trans. Graphics*. Vol. 14, No. 3, pp. 203-232. 1995.
- [7] S. H. Hong, D. Merrick, H. A. D. do Nascimento. The Metro Map Layout Problem. *Information Visualisation 2004*, Vol. 35 of *Conferences in Research and Practice in Information Technology*, ACS. pp. 91-100. 2004.
- [8] K. V. Nesbitt. Getting to More Abstract Places with the Metro Map Metaphor. *Proc. 8th Int. Conf. On Information Visualisation (IV04)*. IEEE, July 2004.
- [9] M. Ovenden, *Metro Maps of the World*, Capital Transport Publishing, England, 2003.
- [10] E. S. Sandvad, K. Grønbaek, L. Sloth, J. L. and Knudsen. A Metro Map Metaphor for Guided Tours on the Web: the Webwise Guided Tour System. *Proc. 10<sup>th</sup> Int. WWW Conference*, 2001. ACM. pp. 326-333.
- [11] J. M. Stott, P. Rodgers. Metro Map Layout Using Multicriteria Optimization. *Pro. 8th Int. Conf. Information Visualisation (IV04)*. IEEE, Jul 2004.