Visualization Techniques for Schedule Comparison

Dandan Huang[1], Melanie Tory[1], Sheryl Staub-French[2], and Rachel Pottinger[2]
[1]University of Victoria, Canada
[2]University of British Columbia, Canada

Abstract

Project schedules are effectively represented by Gantt charts, but comparing multiple versions of a schedule is difficult. To compare versions with current methods, users must search and navigate through multiple large documents, making it difficult to identify differences. We present two novel visualization techniques to support the comparison of Gantt charts. First, we encode two Gantt charts in one view by overlapping them to show differences. Second, we designed an interactive visual technique, the 'TbarView', that allows users to compare multiple schedules within one single view. We evaluated the overlap and TbarView techniques via a user study. The study results showed that our design provided a quick overview of the variances among two or more schedules, and the techniques also improved efficiency by minimizing view switching. Our visual techniques for schedule comparison could be combined with other resource analysis tools to help project teams identify and resolve errors and problems in project schedules.

Categories and Subject Descriptors (according to ACM CCS): H.5.2 [User Interfaces]: Graphical user interfaces,
Evaluation/methodology, H.5.1 [User / Machine Systems]: Human factors, H.4.1 [Office Automation]: Time management (e.g., calendars, schedules), J.2 [Physical Sciences and Engineering]: Engineering.

1. Introduction

Scheduling is an important and complex process for coordinating activities of multiple people in large projects. We focus on scheduling in construction management, which is particularly challenging due to the specialized and distributed nature of these project teams. Construction managers coordinate activities of many diverse work crews, while schedules are constantly changing due to unforeseen conditions such as weather and resource availability.

Project schedules are normally represented as Gantt charts [Tuf83] or activity-on-node network diagrams. In Gantt charts, activities are visualized as bar charts along a timeline based on their start and end dates (see Figure 1). Activity-on-node network diagrams, which are often used for PERT schedules, visualize activities and their logical connections as nodes and links in a network. Current scheduling systems support both schedule representations. Although these approaches are useful for visualizing a single schedule, they are not well suited for comparing and working with different versions of a schedule.

Schedule comparison is a critical issue at different stages during the life-cycle of a construction project:

- **In the design stage:** Many projects are executed with overlap between design and construction. Construction schedules are often created at this stage to help coordinate the design and construction activities, and to ensure that project constraints, such as permitting, are considered. It is important to explore multiple options to plan for potential delays (e.g., permits that are not received in time) and the timing of eventual occupancy.
- **In the planning stage:** Schedules are typically created after the contract is awarded and before the start of construction. Several possible schedules could be proposed initially, considering resource, space, and other project constraints. While working through these various constraints, it is essential that differences among all the potential schedules be clearly communicated.
- **In the execution phase:** During execution, progress is typically checked weekly or monthly. There is generally an overall project schedule but contractors might keep their own execution schedules as well. By comparing the execution schedules with the original plan, progress can be monitored (e.g., activities that are delayed due to poor weather conditions). In addition, if some critical activity is delayed (e.g., structural steel fabrication), the follow-on activities might need to be rescheduled to meet the deadline. To effectively do 'what-if' analysis of alternative schedules, managers need a tool to identify schedule changes and their impact.

All these cases involve schedule comparison. Sometimes only two versions are compared (e.g., comparing actual progress to an original base plan). Other times, more than two may be compared to find an optimal solution (e.g., comparing alternative options for expediting a schedule).

Schedule comparison in current tools is tedious and time consuming. A common solution is to place the schedules side by side and compare the activities one by one. Sometimes users just compare textual dates in spreadsheets, but this is only reasonable for a very small schedule. The Primavera Project Planner, a popular scheduling tool, compares different schedules by simply stacking the time-based bar charts of corresponding activities together into one Gantt chart (Figure 1). This method is space consuming, does not easily scale, and does not help the user to understand the various differences.

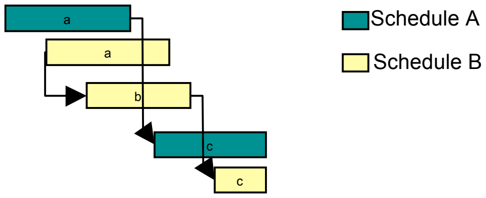


Figure 1: *Stacking approach used in Primavera Project Planner. Two activities are delayed and one activity is added in Schedule B.*

Liston et al. [LFK00] proposed, but did not implement, an *overlap* method, where two related Gantt charts are overlapped in one view. Our discussions with an experienced construction manager suggested that this approach could be very effective. He reported that project managers often manually overlap paper-based Gantt charts (e.g., the contractor's schedule and the owner's schedule), to identify the variances of dates and durations.

We implemented an overlap method as suggested above. We also introduce the *TbarView* technique for comparing multiple schedules. Both were integrated into a prototype scheduling tool. Advantages of our techniques are:

- Encoding two Gantt charts into one view by overlapping, which is more space-efficient than side-by-side views and does not require switching attention between views.
- Providing an overview of all differences between two Gantt charts (side bar view).
- Encoding multiple schedules (normally more than two) in one view to convey overall differences (TbarView).

We conducted a user study to compare user performance between (1) overlap and side-by-side approaches for comparing two Gantt charts, and (2) TbarView and multiple-window approaches for comparing more than two Gantt charts. The study shows that our new scheduling visualization features provide an efficient way to view schedule differences. Such new features in scheduling could not only be applied in construction management but also in project management generally.

2. Related work

2.1. Visualization of schedules

Most current scheduling software represents schedules with the combination of Gantt charts and activity-on-node network diagrams. Aigner et al. [AMT*05] addressed the uncertainty of timing in Gantt charts with PlanningLines, in which the flexibility of activity timing is visualized by two end caps. The bar could be moved between the two end caps, which represented constraints of the timing (Early Start, Early Finish, Late Start, and Late Finish). This idea was implemented in our testbed system but is not our focus.

Network diagrams are useful for identifying the critical path in a schedule (i.e. the set of activities that, if delayed, would delay project completion). Lucko [Luc06] proposed a new layout centered on the critical path. The activities are displayed radially; the row in the center represents the critical path and the distance between an activity and the critical path indicates the critical level of the activity.

We focus on Gantt charts because they are the most common schedule representation in project management. Visual comparison of network diagrams is an interesting topic for future work and may support similar techniques.

2.2. Visualization of comparison

Many patterns can be recognized directly from a side-by-side view of related diagrams [Tuf83]. If there are more than two diagrams to compare, they can be placed in multiple windows [HKD*05]. Variances can be emphasized by connection lines [SB03][Wij08] or highlighting [MGT*03]. For example, TreeJuxtaposer [MGT*03] enabled side-by-side tree comparison. Best corresponding nodes (or sub-trees) were visually marked, and linked navigation ensured that navigation actions in one view propagated to all other views. We implemented a similar side-by-side approach with linked navigation in our scheduling prototype. Other features in TreeJuxtaposer, guaranteed visibility and Focus + Context techniques, may also be helpful for schedule comparison, but have not yet been implemented in our prototype.

One inevitable problem of side-by-side views is the inefficiency of switching attention back and forth between views. An alternative is to combine two diagrams into one. For example, Tu and Shen [TS07] proposed a method to compare two treemaps by combining the information into a "contrast treemap". Corresponding items were mapped into a single item, and color coding was used to emphasize attribute differences. Similarly, some text file comparison programs such as WinDiff [WinDiff] create a combined file and use colors to show additions, deletions, and changes. However, these approaches typically only allow two items to be compared. Our TbarView approach compares multiple schedules, and also shows information about how much activities are expedited or delayed, as opposed to simply identifying 'changes'.

Another way to combine views is direct overlap, where diagrams in different layers can be distinguished by line type or color. Tufte [Tuf93] gave an example depicting the plans of a bridge before and after its collapse, using dotted and solid lines respectively. Isenberg and Carpendale [IC07] presented a visualization in which semi-transparent trees were overlapped and similar/dissimilar nodes were highlighted. In our overlap method, we use line type and color to distinguish layers. We only include layers in parts of the schedule where activities differ.

To our knowledge, side-by-side and overlap methods have not been empirically compared. We conducted a usability experiment to compare these two methods. We also introduce a visualization showing differences among more than two schedules within one view.

3. Visualization design

Most comparison tasks in construction management arise within the context of a single project. For example, a project manager may want to compare two alternative ways to expedite a particular activity, or may want to compare the current progress to the original plan. Therefore, we assume that most activities in the schedules being compared are the same.

Schedules are often represented at different levels of detail to support different tasks. Some tasks may require only an overview (e.g. to determine whether the end date of the project is effected by a change), whereas other tasks may require details such as start and finish dates of a particular activity.

We focus on the novel visualization features we have implemented to support schedule comparison. The new features enable comparison between two Gantt charts using the overlap method, and also comparison of more than two schedules within a single view.

3.1. Design process

We gathered requirement specifications by observing two construction project scheduling meetings and interviewing several construction managers. A paper prototype was refined for a few iterations based on discussions with a senior construction manager. We then implemented a software prototype in Java and revised it based on feedback from a pilot study. This section describes our final design. Variations of this design were created for the user study.

3.2. Overlapped Gantt charts

To support comparison of two schedules, we overlap two Gantt charts and show the variances as 'shadows' (see Figure 2). We also developed a *sidebar* view to provide a visual overview of changes and support efficient navigation (see Figure 3).

In the Gantt charts implemented in our testbed, single activities are visualized by bar charts within two end caps, which represent the constraints of Early & Late Start (Figure 2(d) and (e)) and Early & Late Finish (Figure 2(f) and (g)), similar to PlanningLines [AMT*05]. Similar to the way people overlap two paper-based schedules to examine the variances, the two Gantt charts are overlapped in layers, and distinguished by color and line type (solid or dotted). The two layers overlap perfectly where activities are unchanged, so that 'shadows' emphasize parts that differ. Figure 2 illustrates three examples. The overlap view is displayed in Schedule A's window (left), where the dotted green shadow illustrates how activities are different in Schedule B. The first activity starts and finishes later in Schedule B (Figure 2(a)), and the second activity does not exist in Schedule A (Figure 2(b)). The 'cap' will be emphasized only if the corresponding dates changed (Figure 2(c), where only the left cap is dotted). To reduce visual clutter, the end caps can be optionally hidden.

To access additional details, the user can roll their mouse cursor over an activity. This displays a popup (see Figure3(c)) with activity attributes such as date and duration for *both* schedules, so that details can be directly compared. Attributes differences are highlighted in orange in the pop-up InfoBox.

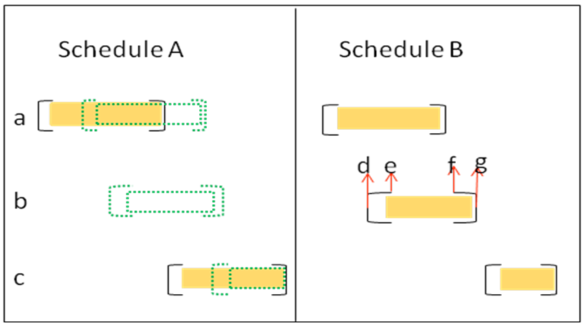


Figure 2: *Overlap view in Schedule A's window. (a) Activity in Schedule B is delayed compared to Schedule A; (b) Activity in Schedule B does not exist in Schedule A; (c) Activity in Schedule B has shorter duration than in Schedule A; (d) Early Start; (e) Late Start; (f) Early Finish; (g) Late Finish.*

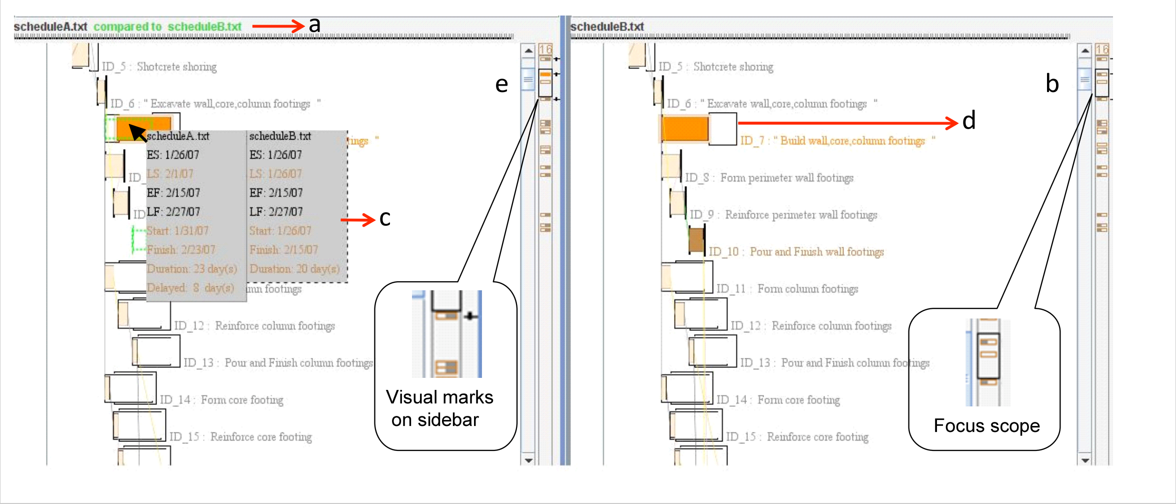
Navigating multiple schedule windows can be cumbersome, particularly when the schedules are large. Users need to zoom in to see details, and then can become lost while scrolling up and down. To alleviate these issues, we designed a sidebar view, which displays an overview of the whole schedule on the right edge of each window (see Figure 3(b) and (e)). Each small rectangle on the sidebar represents an activity that differs between the two schedules being compared. The area covered by a black rectangle (Figure 3(b)) indicates the current scope of the details shown in the schedule window. This enables users to easily tell which part of the schedule is visible.

We designed the sidebar to improve performance when a quick view is needed, e.g., examining whether or not an activity is delayed. If an activity in Schedule A finishes later than in Schedule B, the right half of the small rectangle is shaded on Schedule A's sidebar (Figure 3(e)). Conversely, the left half is shaded if Activity A finishes earlier. A "+" is shown beside the rectangle if the activity takes more time in this schedule (Figure 3(e)), and a "-" is shown if it takes less time. A mouse-over event on the sidebar quickly locates the corresponding activity in the Gantt chart by linked highlighting (Figure 3(d)).

We also implemented some other common features to improve interaction. Our prototype has optional linked navigation so that all windows can scroll and zoom together. Also, corresponding activities can be highlighted in all windows during a mouse-over event.

3.3 TbarView

Users often compare several alternative schedules to one base schedule. For example, a project team might evaluate several alternatives compared to the base to find the best solution to reschedule a project. Overlapping more than two Gantt charts might lead to excessive clutter, and placing them side-by-side is cumbersome and requires users to switch attention among multiple views. Instead, we combine the differences of all the schedules in another visual interactive window – the TbarView.


scheduleA.txt compared to scheduleB.txt
a
scheduleB.txt
ID_5 : Shotcrete shoring
ID_6 : " Excavate wall,core,column footings "
ID_7 : " Build wall,core,column footings "
ID_8 : Form perimeter wall footings
ID_9 : Reinforce perimeter wall footings
ID_10 : Pour and Finish wall footings
ID_11 : Form column footings
ID_12 : Reinforce column footings
ID_13 : Pour and Finish column footings
ID_14 : Form core footing
ID_15 : Reinforce core footing
16
b
c
d
e
Visual marks on sidebar
Focus scope

scheduleA.txt	scheduleB.txt
ES: 1/26/07	ES: 1/26/07
LS: 2/1/07	LS: 1/26/07
EF: 2/15/07	EF: 2/15/07
LF: 2/27/07	LF: 2/27/07
Start: 1/31/07	Start: 1/26/07
Finish: 2/23/07	Finish: 2/15/07
Duration: 23 day(s)	Duration: 20 day(s)
Delayed: 8 day(s)	

Figure 3: *Comparing two schedules: (a) Title shows that schedule A (left) is compared to Schedule B (right); (b) Focus scope shown by black box in sidebar; (c) InfoBox with details of both schedules; (d) Corresponding highlight; (e) Sidebar.*

The TbarView includes a file selection panel (Figure 4(I)), a widget panel (Figure 4(II)), and the Tbar comparison (Figure 4(III)). Users can define their own base and alternatives by drag and drop motions from a separate portfolio window. The portfolio is an interactive panel that stores different schedule versions.

The TbarView conveys 1) difference of End Date of an activity, 2) difference of duration of an activity, and 3) any activity added or deleted. Differences for each alternative schedule are shown by color coding. Our requirements gathering revealed that the typical number of alternatives is small (two to five), so color coding is effective. The same colors are used in the title bars of the corresponding Gantt chart windows (Figure 4(a)), making it easier to visually connect the Gantt chart windows to the TbarView.

The Tbar's backbone is made up of small grey brackets, which represent all the activities in the base, in the same order as in the Gantt chart. Additional marks are added if the activity differs in the alternatives compared to the base. E.g., activity ID61 has a longer duration in the red schedule than in the base, so there is a pair of larger red brackets outside the backbone (Figure4(d)). But if the duration was smaller in the red schedule, the red brackets would be smaller and inside the grey ones. In addition, the size of the red brackets depends on how much the duration of activity ID 61 is longer than in the base. Since that activity takes seven days in the base, nine days in the red schedule, and ten days in the blue schedule, the blue brackets are larger than the red ones (Figure 4(d)). If an activity is deleted in a specific alternative, there is a dotted line across the activity brackets in the corresponding color, and if an activity is missing in the base, the brackets on the backbone will be displayed with dotted lines (not shown).

The Tbar uses bar charts to visualize the difference of end dates (Figure 4(e)). In progress tracking, managers usually care whether certain activities are finished on time. In the Tbar, if the activity of an alternative is finished earlier than in the base, a bar chart will head to the left of the backbone. If it is finished later, the bar chart will head to the right. The length of the bar chart is proportional to the number of days delayed (or expedited). For example, in Figure4(e), the bar charts show that activity ID61 is delayed two days in the red schedule, and three days in the blue schedule compared to the base.

When there are variances in the end dates of more than one alternative for a particular activity, we create a multi-bar cluster (Figure 4(e)). Bars in the cluster partially overlap. When both the duration of an activity and its end date are changed (i.e., there are both colored bars and colored brackets), we move the start of the bars to the outermost bracket. This design choice was made to ensure bars did not hide the brackets. However, the varied start points of the bars confused some users, as described later in the user study results.

We also developed navigation functions to help users link the TbarView and the Gantt charts. Mouse motion on the backbone in the TbarView triggers highlighting of the corresponding activity in the Gantt chart windows (Figure4(c)). To help users see and adjust the visible scope, a dragged box (Figure4(b)) prompts a lockstep update of all the Gantt chart windows, so that is not necessary to adjust the scroll bar or zoom ratio within the Gantt charts.

For example, if the user wants to compare two alternatives (red and blue in Figure 4) with the base to find out the differences of activity ID61, she could select a smaller scope by dragging out a box over the backbone; say the range from activities ID50 to ID70. The Gantt chart windows of the alternatives and the base will adjust their view automatically. She could find activity ID61 by mouse-over. Since both the red and blue brackets are outside the backbone, the activity has longer durations in the alternatives, with blue being the longest. In addition, she could use the bar lengths to see that the activity's finish date is delayed most in the blue schedule. If she needs to find the start date of activity ID61 in the blue schedule, the corresponding activity in the Gantt chart window (with the blue border) is highlighted in bright orange as the mouse moves over the activity on the backbone.

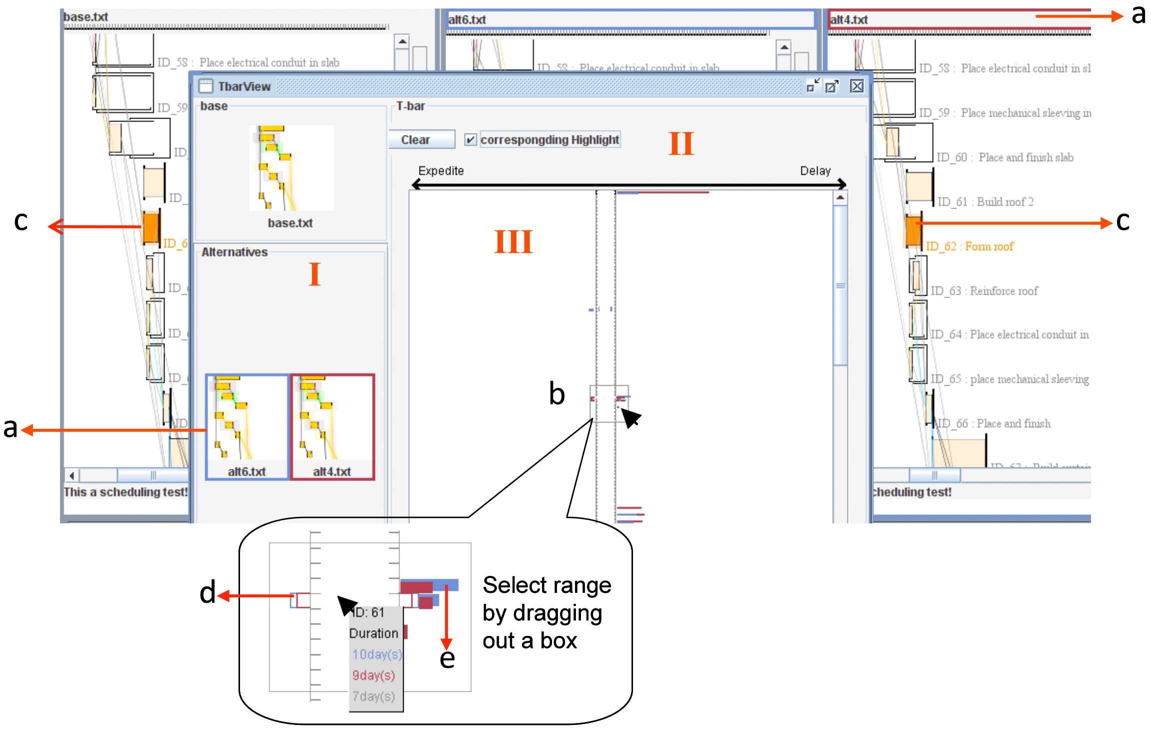

base.txt
alt6.txt
alt4.txt
a
ID_58 : Place electrical conduit in slab
ID_59 : Place mechanical sleeving in
ID_60 : Place and finish slab
ID_61 : Build roof 2
ID_62 : Form roof
ID_63 : Reinforce roof
ID_64 : Place electrical conduit in
ID_65 : place mechanical sleeving
ID_66 : Place and finish
TbarView
base
T-bar
Clear
☑ correspongding Highlight
II
Expedite
Delay
III
Alternatives
I
c
c
a
b
This a scheduling test!
d
ID: 61
Duration
10day(s)
9day(s)
7day(s)
e
Select range by dragging out a box

Figure 4: *Comparing three schedules with TbarView: (I) File section panel; (II) Widget panel; (III) Tbar comparison; (a) Colors to distinguish schedules on Gantt chart title bars and in the file selection panel; (b) Select range by dragging out a box; (c) Corresponding highlight; (d) Duration variances; (e) End Date variances.*

User feedback (see the next section) suggested additional features that could be added to the TbarView. Critical activities could be highlighted on the backbone, and more details of an activity could be shown following a mouse-click. To make more efficient use of the space, parts of Tbar where the activities are unchanged could be collapsed, which could also improve scalability.

4. Evaluation

A two-part user study was conducted to evaluate the overlap and TbarView techniques. We focused only on evaluating our novel comparison related features, and aimed to identify the benefits and the drawbacks of the new features for project scheduling.

We expected to verify the following hypotheses:

H1: Overlap would support better performance than the side-by-side approach for comparing two schedules.

H2: Users would prefer to work on one single view (TbarView) rather than switching between Gantt chart views for comparing more than two schedules.

H3: TbarView would improve performance in capturing a summary of differences among multiple schedules.

We recruited twelve participants (3 female, 9 male), who were senior undergrad and graduate students from Computer Science and Engineering. The participants were asked to complete several comparison oriented tasks, with a 10-minute break between the two parts. The entire procedure took one to two hours including an interview. Sessions were videotaped. All tasks were done on a real building project schedule with minor changes.

4.1. Part 1: overlap vs. side-by-side

In the first part of the study, we studied the participants' performance when using overlap and side-by-side approaches for comparing two versions of a schedule:

- **Overlap**: two Gantt charts could be overlapped, as described in §3.2. The sidebar was available and differences were highlighted in the pop-up InfoBox.
- **Side-by-Side:** two Gantt charts were placed side by side, and differences of corresponding activities were emphasized by highlighting. Sidebar and highlighted differences in the InfoBox were not available.

We chose side-by-side to compare with overlap, because it is the most common approach for making comparisons.

4.1.1. Experiment part 1 procedure

Participants began with a tutorial and warm-up session with a small sample schedule. The moderator was present during the warm-up session, and then observed from a separate room linked by video. Participants worked on twelve tasks independently (six tasks for each interface). They were asked to complete the tasks as fast as they could, and to indicate verbally when they felt confident about the answer. The tasks were comparison oriented, and based on discussions with professional project managers. Two sets of corresponding tasks were created (see Table 1). In Task

2 and Task 6 activity IDs were provided, and in Task 3, Task 4 and Task 5 the ranges of IDs were provided. The complexity of two task sets was kept as similar as possible (e.g., we kept the size of a search range the same but changed the IDs for the range). Order of the interfaces was random, but order of the two task sets was the same across subjects. In this way, any potential differences in difficulty between the task sets should be counterbalanced.

Table 1: *Tasks for comparing two versions of a schedule using Overlap vs. Side-by-Side methods*

Compare Schedule A to Schedule B	
Task1	Is the project finished later in Schedule A?
Task2	Is a specific activity finished later in Schedule A?
Task3	Within a specified range, find out which activity is finished earlier in Schedule A.
Task4	Within a specified range, find out which activity has a longer duration in Schedule A.
Task5	Within a specified range, find out which activity is deleted in Schedule A.
Task6	Compare the attributes of a specific activity.

4.1.2. Part 1 results

We started timing when the participant began to read a task, and stopped when they verbally indicated the answer for the task. Since the counterbalanced design could cancel learning effects and we found that sometimes participants looked back to the task description while performing the task, we take the task reading time into account. Data were discarded if the participant did not complete the task or depended on the moderator to obtain the answer.

Results were analyzed statistically using the paired-samples t-test. Means were estimated with 95% confidence. Results of completion time are shown in Figure 5.

Performances of Task 1 and Task 3 were significantly different between the two techniques, with *p* values (*t*= 3.2, *p*=0.008) and (*t*= 2.7, *p*=0.024) respectively. On average the participants spent less time with the overlap approach, although the values were close in Task 4 and Task 6. Errors occurred when the participant did not complete the task or mainly depended on the moderator to finish the task successfully. Across all participants, there were only three errors (one for overlap and two for side-by-side).

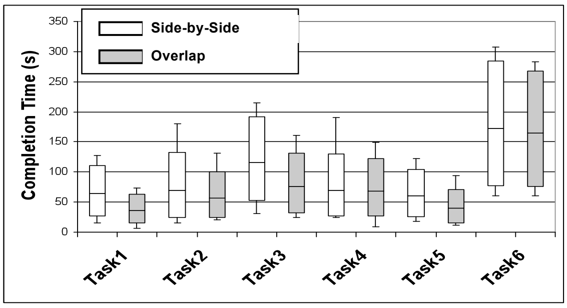


Figure 5: *Task completion time compared between overlap and side-by-side approaches*

4.2. Part 2: TbarView vs. multiple schedule windows

In Part 2, we compared users' performance when using TbarView and multiple-window approaches for comparing more than two versions of a schedule.

4.2.1. Experiment Part 2 procedure

The same participants were asked to compare 4 schedules (one of which was the base). The two interfaces were in random order, but all participants completed the two tasks in the same order. Participants began with a tutorial, but did not perform warm-up tasks since we assumed they learned the scheduling tool from Part 1. They then completed 10 tasks independently (5 for each interface, see Table 2). In Task 3 and Task 5 activity IDs were provided, and in Task 2 and Task 4 the ranges of IDs were provided. Instructions were the same as in the first part (§4.1.1).

Table 2: *Tasks for comparing four versions of a schedule with the TbarView vs. multiple schedule windows*

Task 1	Compare finish dates of the project in the alternatives
Task 2	Within a specified range, find the activity that has the largest delay
Task 3	For a specific activity, compare the finish dates (i.e. the time delayed)
Task 4	Within a specified range, find an activity that has different duration than the base
Task 5	Compare the duration of a specific activity in different alternatives

4.2.2. Part 2 results

Statistical results of completion time are shown in Figure 6. The time measure is the same as the first part (§4.1.2). Task 2 (*t*=2.7, *p*=0.02) and Task 5 (*t*=2.7, *p*=0.02) were significantly different for the two interfaces. In most tasks, the participants finished faster using TbarView. However, in Task 3 the mean of completion time for the TbarView version was slightly higher than for the multiple-window version (but not significant). That is because participants spent substantial time physically switching between the main Gantt chart windows and the TbarView.

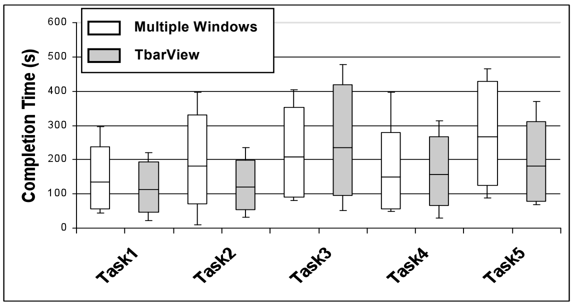


Figure 6: *Task completion time compared between TbarView and Multiple Window approaches*

There were four errors in total, all of which were encountered using the TbarView, especially in Task3. We believe these errors occurred because the participants sometimes forgot about certain features or misunderstood the visualization.

4.3. Domain walkthrough

We also carried out a walkthrough study with four graduate students working in construction management. All had several years of industry experience, and one was experienced in scheduling. They were asked to try the overlap and TbarView techniques after a demonstration. Side-by-side and multiple window approaches were also introduced to them. We then conducted an interview. The whole procedure took 45 minutes to one hour. The domain walkthrough helped us to identify application scenarios and practical issues in construction management.

4.4. Analysis

Overlap vs. side-by-side: Participants performed faster, on average, with the overlap approach in tasks with a binary answer (Tasks 1, 2, 3, and 5). For example, in Task 1 the participants were asked to find out whether the overall project was delayed. Here they could examine the first activity, which serves as the summary of the whole project. The overlap approach benefited their performance in those tasks, because users could capture more information at first glance. They could focus on one schedule window which encoded all the information. Users also liked the linked navigation and thought it could save a lot of their time.

We expected the sidebar to make it easier to quickly answer binary questions. For example, we expected users to take advantage of the "+" / "-"signs on the sidebar to compare the durations of a single activity (Task 4). However, most of the time participants focused on the Gantt chart and ignored the sidebar, so no significant difference was observed for Task 4. Some participants forgot to use the sidebar, and some thought the size of marks on the sidebar were small, so they were more confident with the Gantt charts. Some participants pointed out that information was limited on the sidebar (e.g., only binary information), so they checked the Gantt charts even after getting the answer from the sidebar. Also, the mouse sometimes slipped in the sidebar (which is just beside the scrollbar), causing the focus area to jump accidently. A mouse-click event might work better in this case.

With more practice, users might use the sidebar more frequently, making the benefit more obvious. However, as a project's size grows, the features on the sidebar could become cluttered. We found that the sidebar worked well in small and moderate sized schedules. In a large schedule, Focus + Context techniques on the sidebar could help. Meanwhile, to make the sidebar less cluttered, it could mark only activities on the critical path – those activities that determine the shortest possible time to complete the project. Normally, the number of critical path activities is much smaller than the total number of activities. Some participants also suggested a customized side panel which could display more information beyond the binary marks.

In the overlap approach, differences of activity attributes were highlighted in a pop-up InfoBox. For example, if the durations of the activity in two schedules were different, the text in the InfoBox would be highlighted. We expected this direct comparison to benefit Task 6, but found that it did not have much influence on performance.

TbarView vs. multiple schedule windows: Participants liked the TbarView, and were significantly faster with the TbarView in two tasks. All users preferred to scan the TbarView first for a summary, and then use Gantt Charts for more details if necessary. This suggests that the need to switch cognitive attention between views was reduced by TbarView, as we had hoped. However, because the TbarView was a separate window, we encountered another problem — physical view switching. With one monitor, the TbarView inevitably overlapped with the Gantt charts (see Figure 4), so users had to adjust the positions of windows. This physical switch was irritating to users, and they got frustrated as the switch frequency increased. We expect that if the TbarView was displayed on an extended monitor, or as a panel rather than a separate window, physical view switching would be unnecessary.

Most of the time users liked to focus on one view during the comparison, particularly when they were comparing more than 2 or 3 schedules. Therefore, they expected more information about activities on the TbarView, such as the description and the dates. They also wanted the critical-path activities to be shown on the TbarView. If non-critical activities were delayed several days, it might not make any difference to the whole project. But the project could be delayed a lot even if only one critical activity was delayed. Meanwhile, users expected summaries at the milestone level in addition to the whole project scope.

Although there were few errors, all of them happened with the TbarView. This might be caused by switching between overlapping windows, or the two different representations might cause confusion. Perhaps users need time to transfer knowledge from one form of representation to another totally distinct one. Smooth transfers between different representations of data could reduce such errors.

Some participants were confused by the inconsistency of bar start points in the TbarView. If the activity in an alternative has longer duration than that of the base, its corresponding brackets will be larger and placed outside of the backbone (e.g., Figure 3(d)). If the activity is delayed as well, the bar will shift sideways so that it starts at the outermost bracket (blue bar and bracket in Figure 3(d)). We made this design decision in order to reduce clutter. However, this could cause misunderstanding and confusion when users compare a section of activities in which some bars are shifted but others are not. A third dimension may solve this problem, but could also increase display complexity. A better choice may be to keep the starting point of all bars consistent, but to lower the saturation of the bars so that the brackets can still be seen, or move the brackets to the other end of the corresponding bars.

5. Discussion

Our new scheduling visualization features (overlap and TbarView) improved performance at some tasks, particularly those where a quick overview of differences between schedules was all that was needed. These types of tasks are incredibly common for professional schedulers. Managers typically work with schedules hierarchically; they first look at the summary of the project, then go to the summary of milestones, and finally check individual activities only when it is necessary. What they need is to capture as much information as possible with a quick overview. With the overlap technique they can easily indentify delays by the horizontal shift. The TbarView provides a brief summary of all the schedules being compared. Hence, both of these techniques are suitable for existing work practice and should benefit performance.

We also found the interaction time affected performance. As the users clarified, scheduling is a complex process and the scheduling tool is also complicated with many features (e.g., the 'caps' representing timing flexibility). They reported that sometimes they had difficulty completing the tasks because they were not familiar with the testbed; however, participants believed that productivity with the new features would improve with practice.

In general, users seemed more confident with extra details. In Part 1 (**Overlap vs. side-by-side**) some users checked the detailed attributes of activities even after finding the answer on the sidebar. In Part 2 (**TbarView vs. multiple schedule windows**) they went back to the Gantt charts even though they found the answer on the TbarView. Therefore, providing a fast and convenient way to access details-on-demand could improve users' efficiency.

The major limitation of our current approaches is scalability. In both the TbarView and overlap techniques, details are difficult to see when very large schedules are zoomed out. Implementing guaranteed visibility [MGT*03] would be a useful next step to address this issue.

Users also suggested that comparison tools could help identify errors in a schedule. If the duration of an activity differs between two schedules, this might indicate that one schedule has an error. In addition, users usually want to know the reason for any delays, such as weather conditions. To address this concern, it would be helpful to integrate our schedule comparison features with resource analysis tools.

6. Conclusion and future work

Construction managers need to handle dynamically changing schedules during a project, a process that is not well supported by current tools that focus on editing single schedules. We present two novel visualization techniques to support the comparison of Gantt charts. First, an overlap approach was developed for comparing two Gantt charts, and an overview of differences was shown in a sidebar. Second, we designed an interactive TbarView to facilitate comparison of more than two versions. Results showed that overlap and TbarView techniques efficiently conveyed an overview of the variances between schedules. They also minimized the need for cognitive attention switching.

Our new techniques can be extended in several ways. One important issue for scheduling is dependency, which is the logical relationship between activities in terms of workflow. Example dependencies include finish-to-start (an activity has to be started only after another is finished) and start-to-start (those activities need to be started at the same time). Effectively visualizing dependencies is a common problem for current scheduling tools, and visualizing changes in dependencies is even more difficult. Users might also need to analyze the reasoning behind variances or their further impact. To address these concerns, our visualization techniques could be extended and integrated with other resource analysis tools.

Acknowledgements

We thank Yu-Ling Chang and Alex Merritt for developing the base scheduling system on which our prototype was built. This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

[AMT*05] Aigner, W., Miksch, S., Thurnher, B., Biffl, S.: PlanningLines: Novel Glyphs for Representing Temporal Uncertainties and their Evaluation. In *Proc. Ninth Intl Conf. Information Visualisation (*IV'05*)*, July 2005, 457-463.

[HKD*05] Hao, M., Keim, D., Dayal, U., Schreck, T.: Importance-Driven Visualization Layouts for Large Time Series Data. In Proc. *IEEE Symp. Information Visualization,* (2005), 203–210.

[IC07] Isenberg, P., Carpendale, C.: Interactive Tree Comparison for Co-located Collaborative Information Visualization. *IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis 2007)*, (2007), 1232--1239.

[LFK00] Liston, K., Fischer, M., Kunz, J.: Designing and Evaluating Visualization Techniques for Construction Planning. *Computing in Civil and Building Engineering*, (2000), 1293-1300.

[Luc06] Lucko, G.: An Activity and Arrow Arranging Algorithm for Clarity in Schedule Network Diagrams. In *Proc. Joint Intl Conf. Computing and Decision Making in Civil and Building Engineering*, (2006).

[MGT*03] Munzner, T., Guimbretiere, F., Tasiran, S., Zhang, L., Zhou, Y.: TreeJuxtaposer: Scalable Tree Comparison using Focus+Context with Guaranteed Visibility. *ACM Trans. Graphics (SIGGRAPH 2003)*, 22, 3, (2003), 453-462.

[SB03] Sheth, N., Börner. K.: Treemap, Radial Tree and 3D Tree Visualizations. In *Proc. IEEE InfoVis'03 (Poster Compendium)*, (2003), 128-129.

[TS07] Tu, Y., Shen, H-W.: Visualizing Changes of Hierarchical Data using Treemaps. *IEEE Trans. Visualization and Computer Graphics*, 13, (2007), 1286-1293.

[Tuf83] Tufte, E.R.: *The Visual Display of Quantitative Information*. Cheshire, 1983.

[Tuf93] Tufte, E.R.: *Envisioning Information*. Cheshire, CT: Graphics Press, 1993.

[WinDiff] http://msdn.microsoft.com/en-us/library/aa242739(VS.60).aspx. Accessed March 5, 2009.

[HW08] Holten, D., van Wijk, J.J.: Visual Comparison of Hierarchically Organized Data. In *Proc. Eurographics/ IEEE-VGTC Symp. Visualization,* (2008), 759-766.