Creating a Project Plan

Introduction

Project planning is one of the staples of the engineering profession. With a project plan, the engineer lays out the activities (“tasks”) that must be performed, their relationship to one another and the resources required to accomplish the job at hand. When asked what skills they wish to see in newly hired engineers, most industrial advisors list project management near the top of the list. A project plan is a tool for both organizing work and communicating that work to others.

A project plan is not a schedule. While a schedule can be derived from a project plan, the reverse is not possible. A project plan describes not only when work is to be done, but also in what order that work must take place. The project plan identifies the prerequisites for each task, thereby making clear the work flow from task to task. A complete project plan must have the following essential elements:

1. A set of all tasks required to accomplish the project
2. A complete set of dependencies between the tasks
3. Assignment of required resources (time, money, people, equipment, etc.) to each task
4. Imposition of external constraints (start times, delivery requirements, etc.)

Simple project plans can be put together using pencil and paper. But it is common to use software specifically designed to keep track of the resources and dependencies. At UNC Charlotte, the two principle project management software packages are Microsoft Project and Primavera. Project is available to all students on the Mosaic system and will be used in the examples given here. If you are not familiar with all of the operations described below, the most effective use of this tutorial is to create the example for yourself as you read through the tutorial. That will give you the opportunity to practice the commands discussed as well as experiment some with other options that are available but not explicitly described in the example.

Example 1: Making a sandwich for lunch

Entering and formatting the project

Consider the process of making a sandwich for lunch. This is something that we do automatically. But the process provides an example of many aspects of project planning. In the first iteration, one might make a task list that looks like

1. go to refrigerator
2. get ingredients
3. get bread
4. assemble sandwich
5. eat sandwich

Task 2 depends on task 1, and task 4 depends on tasks 2 and 3. Clearly task 5 depends on task 4. As a starting point, we can assemble the project plan with these five tasks. We will return later and examine
the sufficiency of this task list.

To enter our project in Project, we open the program and begin by setting a title for the project (**File => Properties**) and saving it with an appropriate file name. We then have a choice of several views in which to work. The default view is “Gantt Chart” and will be used as the starting point for this example. The default Gantt chart view has a list of tasks on the left and a schedule chart on the right. We begin by making a list of our tasks. When done, the window should look something like the screenshot below.

At this point we have a task list, but there is no information regarding dependencies. While both task entry and dependency entry can be accomplished in either major view, it is more intuitive in the “Network Diagram” view (and this also helps us learn to use another view). To change views, just select **View => Network Diagram** from the menu bar. The resulting view looks like this...
To establish relationships between the tasks, one can place the cursor on one task block, depress the left mouse button and drag a line the the dependent task. The result is the project network shown below.
If we return to the Gantt Chart view, we will see that the dependencies are shown in two ways: 1) there is a “predecessor” listed for each of tasks 2, 4 and 5, and 2) there are lines in the schedule showing the relationship of tasks. It should be noted that display of the lines between tasks in the schedule chart (Gantt chart) is a display option that can be deselected if desired (**Format => Layout**).

We now have a set of tasks (element 1) and a set of dependencies (element 2). We next need to estimate the times required for each task. This is where your engineering judgment enters the picture. These estimates will be entered in the “Duration” column. My estimates are shown below. Your estimates may differ, and likely depend on factors such as the distance to the refrigerator and how carefully you handle the ingredients, or how many things you have to sort through to find them.

For this particular example, we now find that the timescale in the Gantt chart is inappropriate, and the start and finish times that have been calculated are not very helpful either. To address the schedule timescale, place the cursor in the timescale itself and click the right mouse button. Select “timescale” and adjust the scale so that it displays appropriately. I have chosen to have two tiers, hours and minutes. We have now included the important resource of time (element 3). We will come back later and consider some of the other resources that might be important to include. For now, let's look at including the final element.
It is apparent that this is not a complete project plan for making a sandwich for lunch because it begins at 8:00 am and we are done eating before 9:00 am. That is not my idea of lunch. What we need to add is the external constraints. We will do this by adding two zero-duration tasks that mark our release to lunch and lunch over. While we are at it, we will include another zero-duration task to mark an important milestone in our project, sandwich complete. The following screen shot shows the schedule with these three tasks included. The start time of “release to lunch” has been set to “11/17/09 1:00 pm”, and the finish time of “lunch over” has been set to “11/17/09 1:45 pm” to reflect a 45-minute lunch hour.

We can make several observations in this chart. One is that Project has automatically scheduled tasks to begin as soon as possible, limited only by the duration of predecessors. This provides us with information about how quickly the project could be completed. In this case, we can see that we could complete lunch in 36 minutes. The remaining time until the “deliverable” milestone is referred to as float or slack, and reflects the margin in the project plan. This example has also shown us the three basic kinds of zero-duration tasks, or milestones: specified start time, specified finish time and calculated by project plan.

A second observation is that the layout of the chart does not place the milestones in a visually pleasing location to reflect the flow of the project. We might prefer to make the “Release to lunch” milestone the first line, place the “Sandwich complete” directly after assembling the sandwich and leave “Lunch over” at the end. It should noted that there is considerable variation in how people choose to organize
the display of a Gantt chart. But we will make this change to illustrate some of the abilities and limitations of the software. If we move them, we get the following display:

![Gantt Chart Example](image)

This has the display that we were looking for, but a new problem has been created. In all of our discussion, we referred to the tasks as number 1, 2, etc. When we made the new list, Project renumbered the tasks according to their line number. It did preserve the dependencies, renumbering them, but we still have a disconnect between the text of our documentation and the project plan. How do we get the numbering back for reference? The solution is to use another label for the tasks that is independent of the line number. We can see the options by double-clicking on a task. In the dialog box that comes up, we could select “Custom Fields” and create our own custom identifier. But what we will do here is use another specialized field in the “advanced” tab, and that is the WBS code. WBS stands for “Work Breakdown Structure” and is used to organize tasks in a project plan. In this case, we will simply give separate WBS codes to the tasks corresponding to their original number in our list. At a later time we will return and look at better ways to use the WBS code. For now, we are using it for its basic purpose: to identify tasks (or groups of tasks) in a way that connects to other documentation.
When we have entered the WBS codes for each of the tasks, the next problem is to display it on the Gantt view. We do that by placing the cursor in the “Task Name” label, right clicking and selecting “insert column”. The resulting dialog allows us to select the WBS filed and label it accordingly. The end result follows, where “M” has been inserted for the WBS code of the milestones.

As we look at the table, we can see that the start and stop times are not very meaningful. We need to change the format in which they are displayed. Although we have planned for lunch on the 17th, we are really interested in the times. To make this change, select **Tools => Options** from the menu bar and change the “Date format” in the “View” tab of the resulting dialog box. This results in the following display:

![Gantt Chart Image]
This is great except for one thing: lunch does not start at 1:00, it starts at 12:00 and ends at 12:45. But if we adjust the start time to 12:00 for the “Release to lunch” task, it will still defer going to the refrigerator until 1:00. Why is this? The clue is in the crosshatched area to the left of 1:00 pm on the schedule display. This indicates that the current calendar considers this to be “non-working time” and will automatically defer tasks until working time is available. This is the way that Project keeps track of breaks, weekends, holidays and other “reserved” times. There is a default calendar that is used; you can make your own calendar if you want to include non-standard holidays or other times that are not available for work. In this case, since we are explicitly wanting to work at lunch, we can select a different calendar by selecting Project => Project Information from the menu bar and choosing the 24 hour calendar. The end result is a properly labeled and display project for making a sandwich for lunch. If you change the view back to the Network diagram, you will see all of these changes reflected in that view.

Entering resource constraints

It is now time to take a second look at the actual project logic and think about the processes. Project has scheduled every task to begin as soon as possible. While we can override that logic, it is usually the choice that we will make when we lay out a project plan. When we look at the result of that scheduling, we can see that going to the refrigerator and getting the ingredients happens simultaneously with getting the bread. If we visualize making the sandwich, it is clear that these two things will not
happen simultaneously in practice. While one could simply manually reschedule tasks to avoid conflict, it is more useful to examine the source of the conflict and add that information to the project model in the software. In this case the conflict occurs because there is only one of me and each of the proposed tasks require my full attention. Right now the project has no knowledge of this fact. We remedy that lack by attaching resources to the tasks.

One adds the resource by double-clicking on the “Go to refrigerator” task and selecting the “Resources” tab. A resource called “Sandwich maker” can be created on the blank line and automatically assigned to the task at 100% usage. For the next two tasks, the sandwich maker can be assigned from the drop down list that is available in the table. When these resources are assigned, we get the following display.

While there is still no correction to the schedule, Project is now aware that these tasks require a sandwich maker 100%. We can now see our conflict by selecting View => Resource Graph from the menu bar. After adjusting the timescale for a better display, one will see the overallocation of the sandwich maker resource:

Note that if we had two sandwich makers available, we could modify the resource (using the Resource Sheet view) to have peak units of 200% and it would not have been considered overallocated. What should we do about this overallocation? It is clear that we will need to reschedule some tasks. We can do this by hand, or we can ask Project to try and automatically correct the overallocation. To do the latter, select Tools => Level Resources. We will set the resulting dialog to automatically level, do so on a minute by minute basis and not allow tasks to be split. The dialog settings look like this;
Executing “Level Now” results in a project plan that eliminates the overallocation of the sandwich maker, but shifts the completion of the sandwich to a later time, thereby decreasing the project float.

We now have a fully-formed, resource constrained project for making a sandwich lunch.

**Introducing a design function**

At this point our plan for lunch exhibits a shortcoming that is very frequent in project plans. We have described the fabrication of the sandwich, but neglected its design. The most straightforward way to address this is to add a design task ahead of the fabrication. If we take a few minutes to plan our sandwich before setting out to make it, the plan would look like this
Here we have specified a new resource, the sandwich designer, to clarify functions. Because the designer and maker are different resources, there is no overallocation in the parallel tasks. There are now enough connections between tasks that the logic is not easily seen in the Gantt chart. Switching to the Network Diagram view displays the logic clearly.

The default display would provide more information associated with each task and take up more screen space. The display shown here was created by changing the box format with **Format => Box** and creating a template which displays name only on two lines in 12pt type. This change was made so that the network diagram could be more easily reproduced in this tutorial.

We can now identify a risk associated with this plan. What happens if we design a ham sandwich, arrive at the refrigerator and discover that there is no ham? The present plan comes to a stop. At that point we would have to replan, starting again with the Design Sandwich task. It is this kind of thought that will lead us to a more robust and more detailed plan. In this case we can consider a number of possible sandwich designs before we get to the refrigerator. Then we can make a design selection based on a combination of our personal tastes and available ingredients. This is a classic aspect of the design process: considering design alternatives (referred to as exploring the design space), collecting relevant data, developing a selection criterion and selecting a design. To reflect this, we create these tasks and “wire” them together in the project plan. To allow us to work easily in the Network Diagram view, change the layout using **Format => Layout** from the menu bar, selecting manual placement of the boxes and modifying the connections to direct arrows.
We may then move the boxes around to create a more compact display and add the new tasks using either **Insert => New Task** or the “insert” key on the keyboard.

The next step is to use the mouse to create the dependencies between these tasks. In doing so, we will edit the “Design sandwich” task to be “Select sandwich design”. In addition to adding new connections between the tasks, we will also have to break an existing connection from “Release to lunch” to “Design sandwich”. We do that by double-clicking on the link and then deleting the link. After moving the boxes to make a more clear presentation (we could also return to the layout menu option and set automatic placement), we get the following logic:
We need to add estimates for the new tasks. We can do that in the network view, but it is usually more convenient to make these additions in the Gantt view.

Here several changes have been made in the Gantt view.

- The lines have been rearranged to make the schedule display a bit clearer (note that this will alter the network view if automatic box positioning has been selected).
- The “Get ingredients” task has been reduced by the time required to survey the ingredients on the assumption that a survey was the first part of actually getting them.
- Because determination of the selection criterion can be done in advance of lunch, it has not been connected to the Release to lunch, but has been left with a floating start. By manually
setting a “must finish by” date to 1:00 pm, it can be brought into the schedule display. There are several ways that this might be handled, depending on how one wants to incorporate the task into the plan.

- For the moment, the WBS code for the sandwich design tasks has been set to “D”.