GANTT charts with org-babel

Eric S Fraga

$Revision:1.12$

table of tasks and milestones

The following table describes the tasks and other relevant information for a project. Each line is an *entry* and there are three types of entry allowed in this table:

task an actual task that has a start time, a duration and an end time.

milestone a specific milestone in the project that has a start time alone

date a point in time that will be drawn as a vertical line in the GANTT chart (e.g. start of each year).

Each element of the chart will be annotated with the content of the *label* column of each entry. The first column of the table is ignored but I use it to number the entries. The last column, titled *align*, is used to determine where to place the *activity* text for tasks, whether to the left or right of the bar or, if nothing is specified, centred within the bar itself.

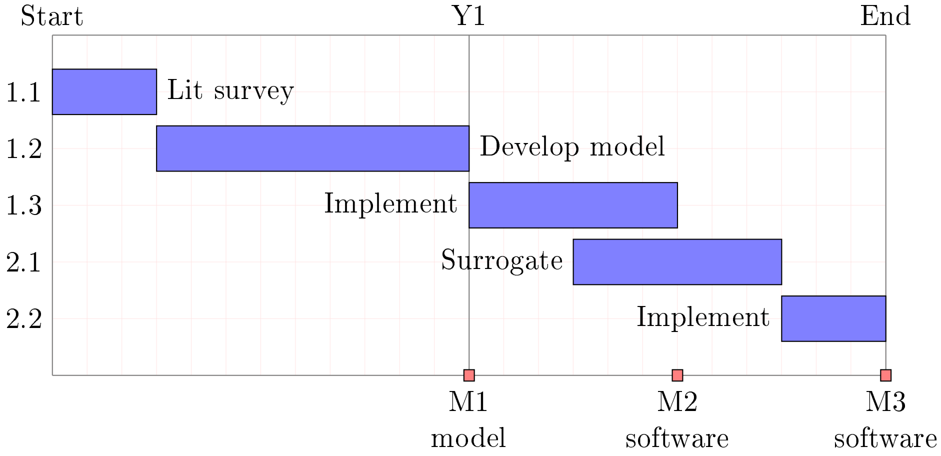
	type	label	activity	depends	start	duration	end	align
1	date	Start			0		0	
2	task	1.1	Lit survey		0	3	3	right
3	task	1.2	Develop model	2	3	9	12	right
4	milestone	M1	model	3	12		12	
5	task	1.3	Implement	3	12	6	18	left
6	date	Y1			12		12	
7	milestone	M2	software	5	18		18	
8	task	2.1	Surrogate	3	15	6	21	left
9	task	2.2	Implement	7	21	3	24	left
10	milestone	M3	software	8	24		24	
11	date	End			24		24	

some latex code for gantt charts

The following LaTeX code is used to generate the actual elements of the GANTT chart. The three colours used, to indicate *tasks*, *milestones* and *date* lines, can be change to suit your own tastes. All the rest of the code is not meant to be changed. Ideally, this code should be in a separate *style* file (see list of actions below) and an environment should be defined. As it stands, the code can only be used once within a document unless the `gantttask` counter is reset manually.

The GANTT chart

The following GANTT chart is generated by the `emacs-lisp` code in the section immediately above this one in the `org` file, a section that has not been exported.



TODO [0/5] outstanding issues or actions

- □ encapsulate latex code within a package
- □ figure out how to make indirect references in org's table
- □ using org-store-link on a section like this one stores the special directive, in square brackets, as part of the link information but, of course, this changes as items are ticked off so the heading changes.
- □ there seems to be some problem with indentation of description lists such as the one in the first section of this document.
- □ although the export procedure prompts for whether the emacs lisp code should be executed, answering yes doesn't actually update the results section. The code is executed (see **Messages** buffer). The updating might not be happening because the code is within a section that is not going to be exported or because there is no

export directive on the code? I should try playing with `call` invocations of babel code.