ChiliProject - Feature # 159: Modularize the calendar

Status:	Open	**Priority:**	Normal
Author:	Felix Schäfer	**Category:**	
Created:	2011-02-08	**Assignee:**	
Updated:	2012-03-12	**Due date:**	

Remote issue URL:

Affected version:

Description: I have some grief with the calendar because it is very very static at the moment.

Example: I've created a plugin to export calendar contents as an ical feed (see "redmine_ical":https://github.com/thegcat/redmine_ical), but it duplicates a lot of the things already in the calendar controller, and still doesn't allow other things than issues and versions to be shown.

My idea would be to create a common interface/module models can include, e.g. @acts_as_cal_provider@ or something like that, which would allow any model to map internal attributes to a common "calendar element" interface, which in turn can be used for every representation of the calendar, i.e. the calendar page and an ical feed.

Thoughts?

History

2011-02-08 10:38 pm - Eric Davis

- Subject changed from Modularize the caldendar to Modularize the calendar

I was thinking the same thing. When I first refactored the calender controller (split from IssuesController) my goal was to go back and add some underlying "model" data to it. Kinda like how the Gantt chart works with Events. I don't know if we want another @acts_as@ for it though...

In a perfect system: the Gantt, Calendar, and Activity would show the exact same data but with different formats. Activity as a stream over time, Gantt as a chart, Calendar as a... well calendar.

2011-02-08 11:20 pm - Felix Schäfer

Eric Davis wrote:
> In a perfect system: the Gantt, Calendar, and Activity would show the exact same data but with different formats. Activity as a stream over time, Gantt as a chart, Calendar as a... well calendar.

I can see how the Gantt and the Calendar _could_ work off of the same data, not the activities though. The current implementation of the Calendar shows due dates of issues and versions, the Gantt needs start and end date and progress on issues and version, so a superset of the former, and the Activity, well, it shows changes on everything, not the current start date of issues (Gantt) or the end date of issues and versions (Gantt, Calendar), nothing of that, so it's a totally different set of data.

Let me take that a little further and imagine you want to have "real" calendar events (not things that happen on a certain day, but things with actually a start datetime and an end daytime. How would that fit into a subset of data you'd want for the Gantt? That's maybe not even something you'd want to show on the Gantt!

As much as I like sharing interfaces/mixins/whatever where possible, different ways to look at information need different filters, and I'm not sure trying to squeeze them all into a common one won't make things more difficult.

2011-02-09 12:00 am - Eric Davis

Felix Schäfer wrote:

> I can see how the Gantt and the Calendar _could_ work off of the same data, not the activities though.

Good point, Activity is different. More of a timestamp and not an event with a time/date range.

> Let me take that a little further and imagine you want to have "real" calendar events (not things that happen on a certain day, but things with actually a start datetime and an end daytime. How would that fit into a subset of data you'd want for the Gantt? That's maybe not even something you'd want to show on the Gantt!

If you think back to the old Gantt, it used to put a little diamond on a date for when a version was due. I think that would work for "real" calendar events. (assume that an hour long event would just show up for the day, drawing hours on the Gantt is a bit out of scope IMO (and not useful)).

> As much as I like sharing interfaces/mixins/whatever where possible, different ways to look at information need different filters, and I'm not sure trying to squeeze them all into a common one won't make things more difficult.

I still think things like Issues and Versions should be represented as events. They can have a date range where they are valid just like "real" calendar events. I think @acts_as_event@ could be extended for this.

Just some rambling ideas... much of this from the massive Gantt refactoring I was working on (but never finished).

2012-03-12 11:19 pm - PaweÅ, O
Definitely Chiliproject needs some nifty calendar integration/modularization so let mi put some thoughts I have:
* There should be some internal calendar server working through caldav. Manual export from CP and import to calendar software is simply unacceptable. The caldav server would allow to connect to it from external calendar viewers as well as internal CP calendar viewer.
* Each project should have its own calendar with its issues and versions. It would be mostly useful for project planning, and deciding which issue goes after which. Though some nifty gannt with drag&drop would be probably more convenient.
* Each person should have its own calendar with Notes/Activities/Timelogs he submitted from all the projects. From that calendar he could manage his timelogs and future (see below) events and tasks.
* There should be possibility to add a starting/ending time for person Notes/Activities/Timelogs (apart from duration). It would be especially useful for developers so they could to see *on their calendars* when and how long they were working.
* There should be functionality added so a future Note/Activity/Timelog event could be added to an issue and to the persons calendar accordingly (with some link to note probably) so again the developers could plan their work in advance easily. Time for future events would be logged after submitting the time spent. Future event could be a note or a meeting request (that would send meeting invitation to all the participants)
* There should be implemented also some effective builtin calendar viewer as proposed in #920 or external one "Agendav":http://agendav.org for displaying calendars within CP