Mozilla Calendar Search

User Interface Proposal
Author: David Dewar, Cognos Incorporated, david.dewar@cognos.com
Status: Draft
Version: 1.0
Contributors: Mike Potter, Gerard Torenvliet
Date: 22-Jan-2002

Design Rationale

The Mozilla Calendar project currently has a search that does not provide enough functionality. This User Interface proposal outlines new directions that the Calendar search could take.

The following are design goals:

- The search should be consistent with the rest of the Mozilla product (in particular, with the other search features)
- Provide a default search that will suit most users needs
- Allow users to customize the search when the defaults are not adequate (e.g. options)
- To provide a simple user interface for finding events (make finding events easy and effective.)
- To optionally provide an advanced user interface for finding events
- To efficiently and effectively allow users to search for events

There are a number of different search user interfaces in Mozilla (browser, email, advanced email search). Each of them has a different set of information to search.

The Internet search will hit a multibillion record database.

Users often receive many emails per day, and a good number of them will keep the email around for years. There are often 1000+ email messages in someone's account. Email messages are also stored hierarchically; this makes it harder to browse. Email typically requires a more advanced search.

Searching calendar events is a different type of search. The number of events is typically less. A smaller scale search strategy will work nicely for finding events. By default search should work on the most important fields (such as title and location, and details and who is invited)

Search Design

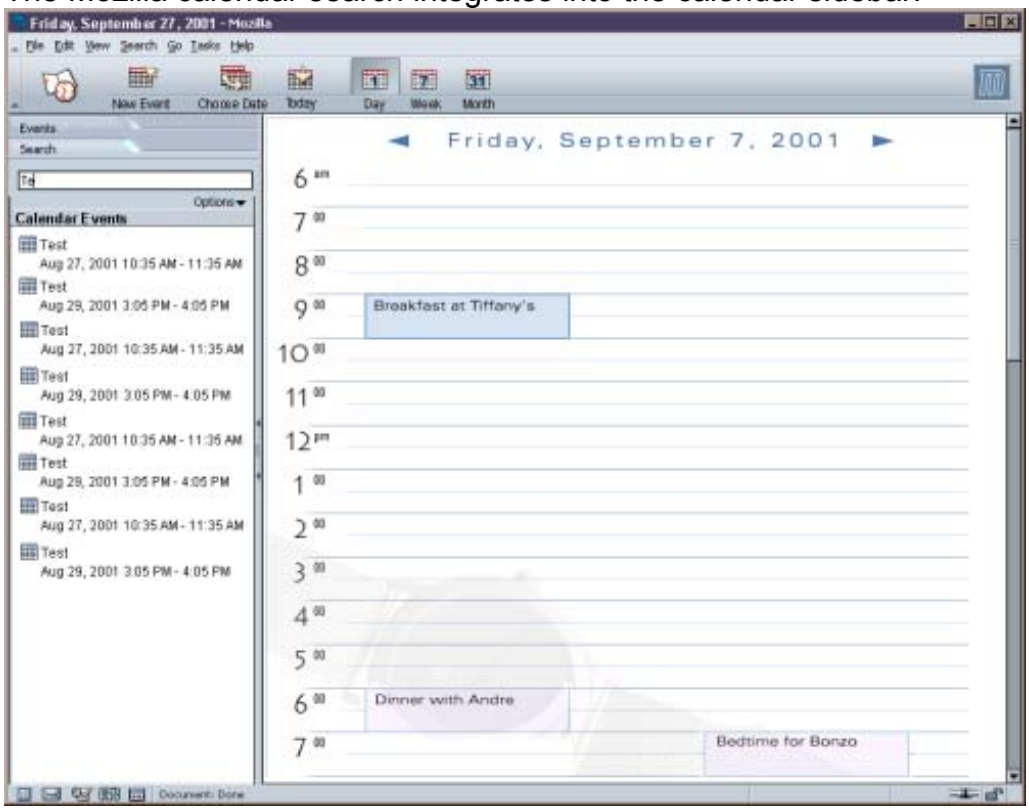
Search comes from the field of information retrieval. The Mozilla Calendar needs a good search, because events are spread out
over time. The search provides an effective way to find events, so that the user does not have to navigate the calendar.

Search Tasks:

- Trying to find a particular event or events
- Trying to see if you are busy at a particular time
- Quickly look up the details of a meeting (location, who, when)…

Search User Interface

The Mozilla calendar search integrates into the calendar sidebar.



The search is put in the sidebar for a variety of reasons:

- The user can navigate a larger number of events quickly, without losing their current context
- The sidebar is consistent with other two-pane products in Mozilla
- The user can hide and show the sidebar as they desire

One drawback is that the space is so small that not much important information can be conveyed. Events don't typically take up much space, so this is probably not an issue.

Specifying a search

The search will match by default against the title, category, location, invitees, time, date and event description. To specify the value to search for, the user will type in the search text box.

Options

The user can click on a ‘button’ called ‘options’ to reveal a set of parameters that they can set to help narrow the search.

Available options:

- What to search (the user can specify a search for all fields or an individual field
- Date range (a set of sliders that can be used to filter the list)
- Amount of detail to return (even title, time, and who is invited, location, category and/or event details)
- Sort (default sort, by category, by date, ascending, descending)

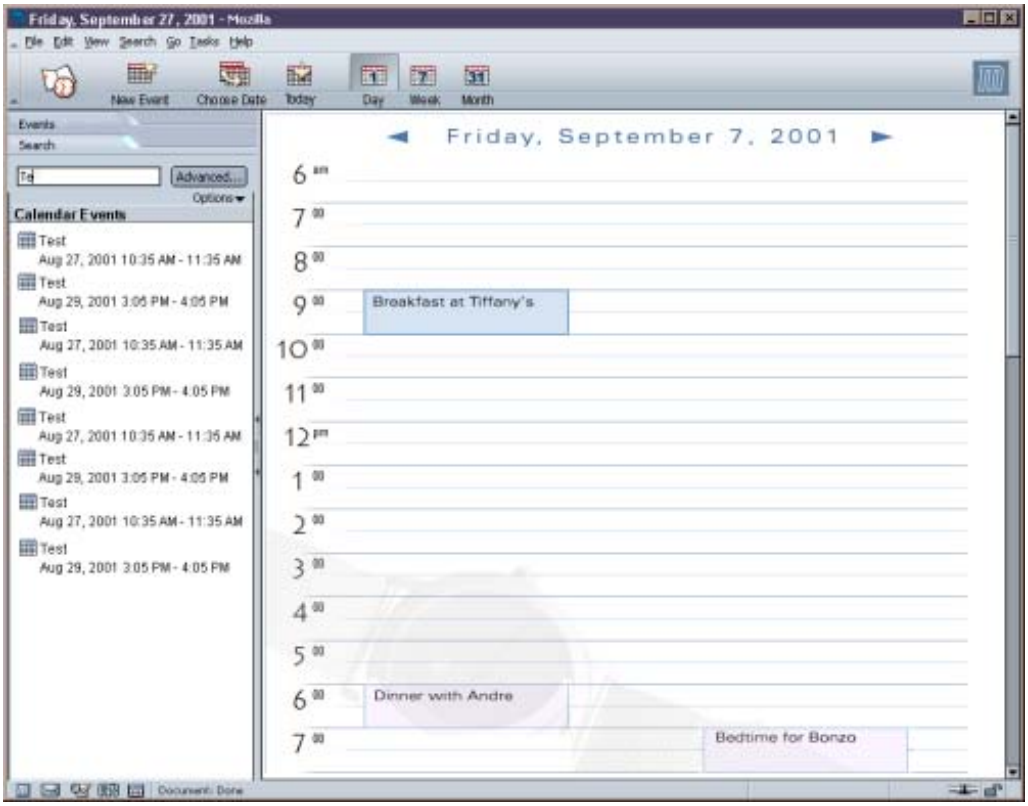
Search as I type

Since the search is in a small scale database, and performance is likely to be good, an Incremental search is a good strategy. By default, the search is a ‘contains’ query, with the first set of results grouped as if it were a ‘starts with’ query.

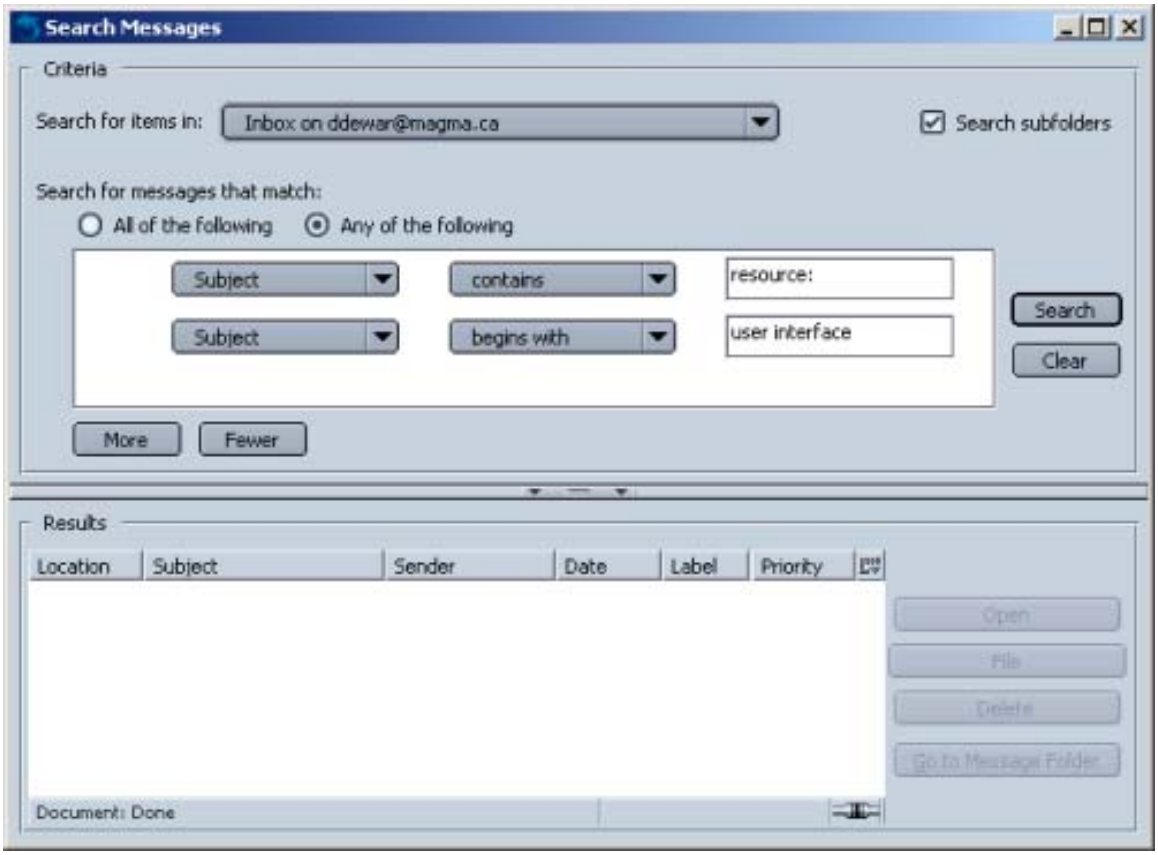
The search is executed according to the following algorithm. When a key is pressed, a timer starts (400 ms). If another key is pressed in that time, then the timer is restarted. After the timer has run its course, then the search is executed and the results are returned. The search is executed regardless of where the time is at after every 4 key presses. This will help reduce the number of queries, but will still feel like a responsive search.

Advanced Search

Optionally, if users require an advanced search capability, a button to launch the advanced search can be provided. This is consistent with the email search (however advanced search makes more sense for the large numbers of email).



The Advanced search dialog for email is shown below. This is probably provides too much power to search through events.



Review the Results

The search should provide a visible indication of state (e.g. searching, done searching). This can be accomplished with the spinning search feedback graphic that is found in other parts of the interface.

When the results of a query return, the matching events show up in a list. The user can navigate through the result list using a scroll bar. Using a list is a good choice because the number of records is likely to be less than 1000 matches. The scrollbar will let the user know where they are in the result set, and provide a familiar way to navigate. It will also have the benefit of saving them from having to navigate between pages of data.

Results should have show:

- the event icon
- the title of the event
- date
- location (optional)
- details (optional)
- who is invited (optional)

Each event should provide the important details of the event when the user hovers over the event. The preview must provide just enough meaningful information to let the user make a sensible decision.

Recurring events should only return one record by default (when everything else matches but date). Recurring events should have a different icon.

If no results are found, then a message should be presented to the user (e.g. ‘Your search did not match any events’).

Single clicking on an event will select it.

Double clicking on an event will open the edit event dialog.

Search Refinement

Once the search is done, it is easy for the user to refine their search by changing their search value and options. The search should immediately re-execute when the user refines the search.