Strategic Interaction and Knowledge Sharing in the KDE

Developer Mailing List

Appendix

Examples of Threaded Messages and Knowledge Sharing
A sample of 5 messages taken from the following threaded discussion:
(http://lists.kde.org/?t=95412579400005&r=1&w=2)

First message: Charles proposed that he was going to combine the "System/Date" and "Time" features together [Recombine]
I'm taking it into my own hands to add a kcontrol applet for setting the date and time. I'm guessing in System/Date and Time would be good. If there's no objection, I'll be starting now.
Charles

Second message: David suggested integrating the existing "kcmclock" feature [Recombine]
What about integrating the existing kcmclock? It's on ftp.kde.org. Re-use what already exists...it saves time and effort. :)
David

Third message: Paul Campbell joined the discussion and drew attention to the existence of various options [Reuse]
Yup - I was planning on doing this (I'd ended up owning it) but hadn't got around to it yet - I've passed it to Charles to use.

The interesting issues are things like time zone support…and how do you do it portably (every other distro or Unix variant has a different way of saving the system time to the hardware clock :-( )
Paul Campbell

Fourth message: in response to the David's suggestion, Dep suggested further feature improvement [Recombine]
kcmclock is really cool, though it could use one improvement: a button that invokes su root and prompts for password, so it's a simple matter to set the system clock from userspace if the user has the privileges. (in the best of all worlds, this would include an option to get the time from the usno or mit, if online at the time, ala the old rdate.)
Dep

Eleventh message (only part of the code is shown here): Paul revealed his patch of code [Reveal]

I wrote a quick & dirty date program a while back before I found out there was one already - it did something like (in 1.1):

```
struct tm t;
t.tm_sec = s;
t.tm_min = m;
t.tm_hour = h;
t.tm_year = today.year()-1900;
t.tm_mon = today.month()-1;
t.tm_mday = today.day();
time_t tm = mktime(&t);
      .
      .
      .
t.tm_mday,
t.tm_hour,
t.tm_min,
t.tm_year+1900,
t.tm_sec);
::system(buff);
}  else {
::system("hwclock --systohc");}
```

Tacky but it works - however it doesn't address the time it takes to type the root password problem - really you need to use your own program instead of 'date' to set the time and to pass a delta time to it.
Paul

Derivation of Conversational Interactivity

The structure and organization of threaded messages resembles a tree-like structure with branches representing the interchanges of ideas, arguments, information, knowledge and so forth. A discussion that branches out more denotes more conversational interactivity than one that is linear. Such structures can be explored by examining the seemingly chaotic branching behaviour in the form of a tree diagram. A classical chaotic system follows the rule that initially close trajectories are separated exponentially fast with reference to time (Gutzwiller 1990). Transformed to suit a discrete system (in this case the tree diagrams representing the discussion), exponential proliferation describes the increment or decrement of the number of nodes on a particular level of the tree diagram with reference to the label number assigned to the particular level (Gutzwiller 1990).

By separating each tree diagram into levels and creating a sequence of the number of nodes at each ascending level, a geometrical series with an average expanding ratio r was obtained. A tree diagram representing a discussion does not necessarily expand for the whole duration of the discussion. More naturally different aspects of the discussion would die out or be resolved at different levels. This suggests that looking at the number of nodes at the last level of the discussion could result in losing valuable information of the diagram's spread. Instead, here the assumption was that the diagram would follow a symmetrical prototype, where the maximum number of nodes would be located at the middle of the discussion. In view of this, "α" is defined as the average ratio of the geometrical series up to the middle level where the

tree diagram would reach its maximum spread. Therefore, $\alpha = \beta^{\frac{1}{\kappa}}$ where κ is the number of levels at the middle of the geometrical series, and β is the maximum number of nodes at any level; alternatively, $\alpha = \beta^{\frac{2}{\nu+1}}$, where ν is the number of levels of the tree diagram.

Here the measure of topological entropy (which is very commonly used in chaotic systems) was applied to denote the intensity of the branching behaviour observed with the threaded messages. Notated as h_t topological entropy relates to "α", in the sense that $\alpha = e^{h_t}$, and therefore $h_t = \ln \alpha$.

Based on this formula, a linear discussion where β =1 would give an $h_t = 0$. However this would also be the case for a question or message posted that did not receive any replies, i.e. for ν = 1, $h_t = 0$. Therefore, in order to differentiate between a single message post, and a linear discussion, the topological entropy of a single message (with no replies) is defined as 0, and the distribution is translated one step right, i.e. $h_t' = h_t + 1$.

For ν = 1, $h_t = 0$; and for ν > 1,

$$h_t' = (\ln \beta^{\frac{2}{\nu+1}}) + 1 \qquad \text{or} \qquad h_t' = (\frac{2}{\nu+1} \ln \beta) + 1 \qquad \textit{Equation} (1)$$

where ν = number of levels of each discussion
β = maximum number of nodes at any level

Equation (1) effectively captures the variation of replies to each message. When there is only one reply, the topological entropy decreases. Furthermore, the length of the discussion bears no relationship with the measure. For example, a long but linear discussion will have an entropy measure of 1, but a shorter discussion with three levels and four messages on the middle (or second level) will have higher topological entropy. For the purpose of this study, equation (1) was used to provide an estimation of conversational interactivity. To test the present measure of conversational interactivity bears no resemblance to the traditional measure based on length, the intercorrelation between the two was inspected, and found that the Pearson's coefficient was not significant with r = 0.16, p = 0.075.

Calculations of *Participation Inequality*, *Conversational Interactivity* and *Cross-Thread Connectivity*

The diagram below shows six developers (A to F) contributed to three threads (T1 to T3). Take T2 for example, the developer B posted twice, and developers C, D, E and F only once. To compute the measure of participation inequality, the following Gini Coefficient (G) equation is used:

$$G = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} |x_i - x_j|}{2n^2 \mu}$$

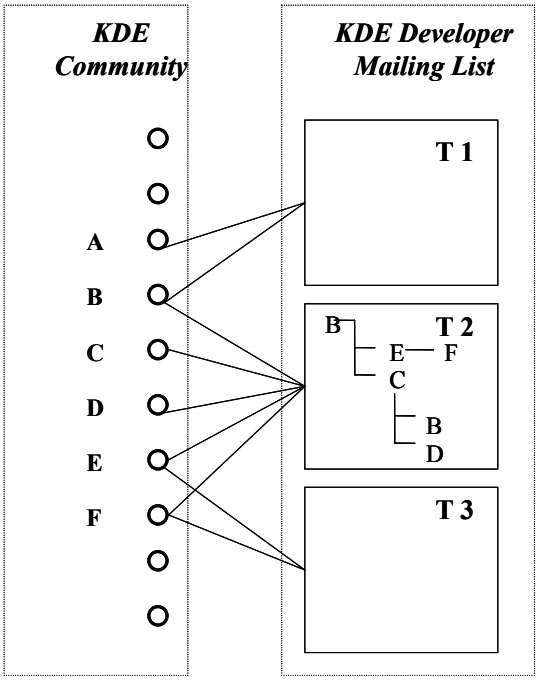
where n = all possible pairs of developers
μ = mean of the postings within a thread

Also, regarding the measure of conversational interactivity, the following formula is used:

$$h_t' = (\frac{2}{\nu+1} \ln \beta) + 1$$

where ν = number of levels in a thread
β = maximum number of nodes at any level

Lastly, for the measure of cross-thread connectivity, the number of common developers is first determined. As shown in the diagram below, T2 and T3 are linked via two common developers E and F. A value of 2 is then entered into the corresponding cell of an n-by-n square matrix, where n = number of threads. This process is repeated across all the possible combinations of threads to populate the square matrix. The square matrix can then be analyzed with any standard network analysis packages (e.g. Ucinet). Here the in-degree measure is used to index the cross-thread connectivity.



T2: Participation Inequality = 0.01

$$= \frac{|B\text{-}C|+|B\text{-}D|+|B\text{-}E|+|B\text{-}F|+|C\text{-}D|+|C\text{-}E|+\ldots+|E\text{-}F|}{2n^2\mu}$$

$$= \frac{|2\text{-}1|+|2\text{-}1|+|2\text{-}1|+|2\text{-}1|+|1\text{-}1|+|1\text{-}1|+\ldots+|1\text{-}1|}{2(20)^2(1.2)}$$

T2: Conversational Interactivity = 1.35

$\nu = 3,\ \beta = 2,\ h_t' = 1.35$

T2: Cross-Thread Connectivity = 1.00

	T1	T2	T3	In-degree
T1		1		0.5
T2	1		2	1
T3		2		0.5