Discuss Enhancements Requirements

Stanley Zanarotti

22 July 1992

Introduction

This document identifies a set of requirements for enhancing the Discuss networked electronic conferencing system. As directed by the Opportunity Evaluation, this document describes a number of desirable features and improvements to Discuss. It provides a list of desired features, an analysis of their costs and implications, and recommendations of what to implement.

This list was gathered from two sources: the archive of previous Discuss suggestions and bug reports (the `discuss` discuss meeting), and interviews with a number of leading Discuss users in DCNS, CSS, and ACS. These users were chosen because of their heavy use of Discuss, and their suggestions in the past. The interviews were used to learn which Discuss interface they used, and provided emphasis to those features that would help their job the most.

The interviews and resulting features are biased towards Information Systems' internal use of Discuss. Most of these users have used Discuss for years, and are quite familiar with its user interface. Although the faculty liaisons and Help Desk considered their respective customers, most of the feedback centered around frequent users. If we had wanted to identify requirements to substantially improve the user interfaces or reduce the learning curve, we would have conducted interviews with more casual users.

The archive and interviews produced a total of 63 suggestions and bug fixes to Discuss, of varying importance and levels of difficulty. Implementing all of them would involve a substantial effort and consume more resources than is practical. We therefore must choose a set of requirements based on importance and expected effort. In this document we divide the suggestions into 3 categories:

1. those that should be implemented in this round of improvements,
2. those that may be implemented if there are sufficient resources,
3. those that will be left to other projects.

In order to choose between these categories, we need to have a rough idea of the available development resources. We assume two to six months of Stan Zanarotti's elapsed time, with 4 to 5 days of billable time per month.

Compatibility requirements

Discuss is a running system with an installed base of meetings, Discuss servers and client programs. We must assume that any upgrades will be installed at varying times on the MIT network, and may even be backed out if problems occur. Different versions of Discuss software will need to interoperate smoothly, otherwise the amount of effort to install a new version would greatly increase. Therefore we have the following set of compatibility requirements:

1. An old Discuss client should be able to work with a new Discuss server without problems.
2. A new Discuss client should be able to work with an old Discuss server without problems.
3. A new Discuss server should be able to use an old meeting automatically.
4. There must be some mechanism to revert a Discuss meeting modified by a new server so it can be used by an old Discuss server.
5. If an old server and a new server cannot interoperate on a meeting format, it should be clearly documented how the servers interact.

Features to be implemented

Named flags on transactions

Currently the Discuss user interface provides access to only one flag per transaction, although the server is capable of storing 15 flags. For meetings that track tasks, these additional flags could be used to record additional status information. The user interface should allow users to set multiple flags on a transaction. Since it would be difficult for users to keep track of flag numbers, these flags should somehow be named to reflect their semantic meanings. These meanings will vary from meeting to meeting, so the meeting will need to keep track of the flag names.

To implement this feature, the Discuss server will need to be changed to store the mapping between flags and their names. The protocol will need new operations to set and retrieve flag names.

The user interface will need to be altered to display these additional flags on transactions, and new commands will be needed to set flag names and flag bits. New transaction specifiers would be needed to select transactions by what flags have been set or not set. (4-5 days)

Searching of meetings on server

Currently the only way of searching a meeting for some given text is either by printing out all transactions and using the search capabilities built into `more`, or by using `dsgrep`, a separate utility program developed by Lucien Van Elsen to search the OLC logs. Both methods are rather slow, because the text of all searched transactions is sent from the server to the client to be searched.

A more efficient way of searching would be to perform the search on the server, and send the transaction numbers of matching transactions back to the client. Although full regular expressions, such as those used in `grep` or `emacs`, would be the most general way of specifying searches, a simpler but useful method would be a series of keywords, with either AND or OR conjuctions between them, that would be searched in a case insensitive manner.

To implement this feature, the Discuss protocol would need a new search request to send the keywords to the server, and receive the transaction numbers back. The server would need to implement this feature, being careful about not locking the meeting too long for new transactions. The tty user interface would need to be changed to make use of this feature, with some syntax in the transaction specifiers to select a set of transactions based on keywords. (4-5 days)

Rechaining transactions

If a Discuss transaction is accidentally chained to the wrong transaction, or starts a new chain when it should actually be chained to a previous transaction, there is no way to fix up the situation.

The Discuss server should allow transactions to be rechained after they are entered. A new request, rechain, should be added to the protocol to implement this feature. (1 day)

Use Moira to generate Discuss acls

Discuss Access Control Lists (ACLs) only allow a list of individuals to be given access, and have no way of setting access for Moira groups. For meetings where an organizational group of individuals need access, such as the OLC logs, the meeting acls need to be updated whenever the group changes, tracking changes that are already made in the Moira database. If Discuss and Moira were tied closer together, it would reduce the administrative work for maintaining Discuss acls.

There are two ways of implementing this feature. One way would be to have Moira generate all Discuss acls. Since Discuss acls specify not only individuals, but the access these individuals have, Moira would then need to have some way to encode the "acdorsw" capabilities, most likely by having 7 lists for a single Discuss acl. With 497 meetings, this would be add load to the Moira database. This approach would also go against the distributed philosophy behind Discuss, since creating Moira lists is tightly controlled in the Athena environment.

The second approach is to use Moira to generate group lists, and have Discuss acls refer to these group lists to give access to groups. The acl syntax would be changed to have a way to support a group name, such as ":consultants". The group membership would be stored in a file or set of files on the Discuss server machine, and the Discuss server would refer to these files to check if a principal is in a group.

We choose the second approach because it does not add significant load to the Moira database, and it gives meeting chairmen more autonomy to control access to their meeting.

If an individual is specified in a Discuss acl, the access specified for the individual overrides the access due to group membership.

If an individual is not specifically listed in a Discuss acl, but belongs to several groups in a Discuss acl, the access will be that of the first group listed.

Some thought will need to be put into deciding what groups Moira should be generated for a Discuss server. A simple approach would be to generate all Moira groups, but this would be slower to generate, slower to access, and take more disk space. Since only a few groups would be used in Discuss meeting acls, it might be useful to have Moira identify these groups, and only generate these groups for a Discuss server. This depends on how difficult it is to add an attribute to Moira groups. (1 day)

Hesiod to add meetings

To add a new Discuss meeting, a user either needs to know the pathname for the meeting and what Discuss server the meeting is on, or find a meeting announcement in one of the announcement meetings. Both approaches are difficult for new users to deal with, especially when a friend told them to "add the ijr meeting." Using Hesiod to maintain a single-level namespace would simplify this process, removing a significant barrier for new users.

There are three possible ways of using Hesiod in Discuss. The simplest and easiest to implement would be to use Hesiod when adding meetings. The user could specify a simple name, which would be looked up in Hesiod to find the server and pathname, which are then added to the user's `.meetings` file.

The second approach would be to store simple names in the `.meetings` file, and use Hesiod to figure out the server names and pathnames dynamically. This allows meetings to migrate between servers without notice, but slows down the process of checking meetings, requiring a Hesiod request for each listed meeting. This approach may also force all meetings to be listed in Hesiod, even private meetings. This approach requires more work, since the `.meetings` format will have to be changed or updated.

The third approach would be to use Hesiod when adding meetings, like the first approach, but also to use Hesiod when an error occurs locating a Discuss meeting. If a Discuss server is down, or the meeting disappears on a server, the client would use Hesiod to see if the meeting is moved, and update the location automatically. This has the advantage of allowing arbitrary meeting moves for meetings listed in Hesiod, without the overhead of repeated Hesiod requests. The server and pathname in the `.meetings` file is a "hint" of the meeting location.

We choose the first approach, since it requires the least amount of effort to implement, while significantly improving the method of adding meetings. If resources are available, we can work on implementing parts of the third approach. (2 days)

Faster listings

When listing transactions, retrieving the information about each transaction involves a network round-trip between the server and the client. This slows down listings.

Adding a list range request to the Discuss protocol will batch requests for transaction information, and speed transaction listings. The request should be added carefully so that it does not excessively lock the meeting, or cause listings to be jerky. User interface research shows that users prefer consistent response time over variable response times. (1/2 day)

Forwarding transactions

Discuss has no provision for forwarding transactions between meetings, except for writing out the text into a file, and manually entering it in another meeting. The Discuss client should have some way of easily forwarding a transaction, or a chain of transactions, between meetings. (1/4 day)

In-reply-to not bounce if no transaction

The `dsmail` program currently fails and bounces a message if the `In-reply-to:` field refers to a transaction that does not exist. This occurs when a message is sent to multiple Discuss meetings, with a transaction number in the `In-reply-to:` field for one of the meetings.

This bug should be fixed. (1/8 day)

In-reply-to syntax include meeting name

If a message is sent to multiple Discuss meetings, there is no way to specify what meeting the `In-reply-to:` field refers to. If a transaction is listed, the entered transaction is either mis-chained in one meeting or bounces (see above).

`dsmail` should be fixed up to accept meeting names in the `In-reply-to:` field. The syntax should be human readable, such as "[0100] in Discuss_development". The meeting long name or short name should be acceptable, and multiple transaction numbers for different meetings should be accepted.

The emacs interface should be changed to generate this syntax. (1/4 day)

Chaining by subject for mail

`dsmail` only chains new messages if they contain a transaction number in the `In-reply-to:` field. If the message senders are not aware of the Discuss archive, then no chaining will occur.

There should be an option for `dsmail` to chain transactins based on subject. Mail messages on the same topic tend to keep the same `Subject:` field, and dsmail should chain transactions if it finds a subject match in the last few transactions of a meeting. A transaction number in the `In-reply-to:` will override this option, and messages without subjects will not be chained. (1/4 day)

Easier mail feeds

To set up a mail feed into a Discuss meeting, the aliases file for the Discuss server currently needs to be changed to add the pipe into `dsmail`.

It would be nice if the sendmail config file on the Discuss server supported a syntax to feed Discuss meetings. Mail sent to a certain type of address would automatically be fed into the appropriate meeting using default options to dsmail. Users could then add this address string to mailing lists, without having to change the aliases files on the Discuss servers. A possible syntax for the addresses would be: `ls.discuss@picasso.mit.edu` would enter a transaction into the meeting `/usr/spool/discuss/ls` on `picasso.mit.edu`. (1/4 day)

`dsmail` not truncate mail headers

Currently `dsmail` only enters the first line on a mail header into the discuss meeting. `To:` fields or `Cc:` fields that span multiple lines are truncated, so that readers of the Discuss meeting do not have a complete list of recipients. This is a slightly annoying bug.

`dsmail` should be fixed to display headers that span multiple lines.

Follow `.meetings` symlinks

Because of the way Discuss updates the `.meetings` file, the `.meetings` file cannot be a symlink to another filename. Although a user can specify a real destination with the MEETINGS environment variable, this works differently than most other dot-files in the Athena environment.

Discuss should follow symlinks when updating the `.meetings` file. (1/8 day)

Fix delay with finding current transaction

If a user has not gone to a Discuss meeting for a long time, the highest transaction he read may be deleted. If this is in a long run of deleted transactions, it may take a long time to find the next non-deleted transaction.

This delay should be reduced, by being smarter when searching for the next transaction.

Fix `dsc_setup` when `.meetings` is not writable

If the user's `.meetings` file is not writable for some reason, like losing authentication, the open fails and the user is asked to run `dsc_setup`.

The error message and behavior should reflect the lack of access. (1/8 day)

Quiet meetings

Currently, all meetings in the user's `.meetings` file are checked when doing a `check_meetings`. Users sometimes read meetings that they do not want to always keep current on. Deleting and adding these meetings is inconvenient.

Discuss should allow users to keep meetings in the `.meetings` file that are not checked by `check_meetings`. These will be called "quiet" meetings, and marked by a "q" in meeting lists. The flags field in the `.meetings` will store this flag. (1/2 - 1 day)

`list_acl username`

To list the access of a single user, the user must be specified by username and realm name. This is hard to type.

The realm name should default to the local realm when listing the access of a user. (1/8 day)

Short names for `-flag_set`, `-flag_reset`

The -flag_set and -flag_reset options used for selecting transactions are hard to type, with both a "-" and "_" character.

These options should have short names without underscores. (1/8 day)

Transactions as a short help name for transaction_specifiers

To get help on how to specify transactions in the tty client, you must type "help transaction_specifiers" or "help specifiers". This is difficult for users to remembers.

"help transactions" should work.

Listing transactions by date, etc

Currently there is no way to select transactions according to criteria other than flag states, transaction numbers, or chains.

Transactions should be specified by additional criteria, like authors or dates. Some of this requirement will be addressed by previous requirements, like searching of meetings on server, or named flags. (2 days)

`print -by_chain`

When printing or writing transactions, there is no way to print out a range of transactions organized by chains. This alternate way of displaying transactions would make it easier to follow the threads of conversation in a Discuss meeting.

This feature should be added to the Discuss tty interface. When the user specifies the `-by_chain` option to the `print` or `write` commands, the command should print out an entire chain it encounters one. (1 1/2 days)

Announcing meetings on someone else's behalf

When user accounts creates a meeting on someone else's behalf, they usually announce the meeting in one of the public announcement meetings. Occassionally the accounts consultant gets questions about these meetings, even though the meeting is chaired by somebody else.

There should be a way to explicitly announce a meeting on someone else's behalf, and the meeting announcement should contain the name of this individual, so that questions about the meeting are directed to the correct place. (1/2 day)

Removing one's own chairman access

There is a safety feature in Discuss that prevents meeting chairmen from removing their own chairman access to the meeting. This is to prevent situations where no one is the chairman of the meeting. When accounts consultants create a private meeting, they want to remove their own chairman access to it, but cannot because of this safety feature.

This safety feature should be changed to allow chairmen to remove their own access, providing that somebody else would still have chairman access. This command should be confirmed before executing. (1/4 day)

Features to be implemented if there are resources

Host byte-order independent meetings

Meetings are currently stored in a byte-order dependent manner. This means that meetings cannot be moved easily between Discuss servers with different byte ordering. Also, some users want to have insecure Discuss meetings stored in their own file space, but the byte-order dependency is a big obstacle for this.

The Discuss servers should be able to deal with meetings of either byte-order. The Discuss meetings have version numbers stored in them, and it should be easy to detect byte-swapped version numbers. Having meetings stored in user local space will not be implemented for the moment, due to the security issues involved. (2 days)

Run external program when transaction gets entered

There is no way for a Discuss meeting to feed a mailing list. It would be convenient for a Discuss server to be able to automatically run a program when a transaction is entered, which could then send it out on a mailing list, or update another database. For the mailing list approach to work, `dsmail` needs to be able to detect this situation and not enter this mailed transaction again, otherwise a mail loop will occur. (1 day)

Better handling of over-quota condition

If the user is over quota when updating the `.meetings`, the Discuss client will not properly inform of this fact, but will simply not update the `.meetings` file.

The Discuss client should report the over-quota condition to the user. (1/2 day)

Ignore blank lines in `.meetings` files

The `.meetings` file is an ASCII file that users occassionally edit to change the order of meetings, or manually add meetings. If the user accidentally adds an blank line to the file, the Discuss client will complain about the file format and not work.

The Discuss client should be more tolerant of blank lines. (1/8 day)

Have a way to tell who set the flag on a transaction

There is currently no way to tell who set the flag on a transaction. For task meetings which make heavy use of flags, this is a piece information that users would like to know.

There should be a convenient way to record this information. Because of the meeting format, it is not easy to add arbitrary amounts of information to existing transactions. Instead, the command language for turning on and off flags should be extended to automatically add a log message for who changed the flag. Then users of tracking meetings can use this language to easily log these changes. (1/4 day)

Better "rn" interface

Some users prefer single-character command interfaces along the lines of the news readers like `rn`.

Although creating an entire new tty interface using single-character commands would be too much work, some benefit would come from extending the current "rn" mode for Discuss. This is a command that automatically walks through changed meetings, with "rn"-like key sequences. Adding a few requests would not be that difficult. (1/2 day)

Better error indications of bad server setups

When setting up a new Discuss server, the administrator needs to be careful about directory ownership and acl files, otherwise meeting creation will fail with an "Invalid access" error. This error is ambiguous and indicates several possible conditions.

The Discuss server should be changed to give more precise information about what the problem is. This would make it easier for new administrators to set up Discuss servers. (1/4 - 1/2 day)

`move_meeting` command

Currently, the only way of changing the order of meetings is to manually edit the **`.meetings`** file. If the user makes an error, Discuss will not be able to read the file.

There should be a move_meeting command to move individual meetings to specific places in the **`.meetings`** file, so that users can easily control the order of meetings. (1/2 day)

`sort_meetings` command

Users may wish to sort their meetings by either short name or long name. The Discuss user interface should have an easy way of sorting meetings. (3/4 day)

Change OLC to use author field

The OLC logs currently store the name of the user in the transaction Subject field. If this were stored in the soft author field, then there would be more screen space to display the description of the question.

Once Discuss allows searching of the soft author field, the OLC logger should be changed to use the soft author field. (1/2 day)

`announce_meeting` that automatically enters transaction

When reannouncing a meeting, the text of the announcement is usually stored in the first transaction of the meeting, or in a previous announcement. (1/2 day)

The **`announce_meeting`** command should allow the text of the announcement come from a transaction. If the transaction is an announcement, the announcement text should be stripped off. (1/2 day)

Fix up `write` command

The syntax of the **`write`** command in the tty interface is confusing, especially when the file name is omitted. The syntax should be fixed up to be cleared. (1/2 day)

Ability to find out space used by meeting

There is no easy way for a chairman to find the amount of space a Discuss meeting is using, or how much space an expunge would free up. Space management on Discuss servers would be easier if the chairmen could keep better track of these numbers.

The Discuss server should provide a way of presenting this information, either by keeping track of space as transactions are added and deleted, or by providing operations that calculate these numbers. (1/2 - 1 day)

Man pages for discuss server programs

There is currently no man pages for some of the Discuss server utility programs, such as **`mkds`**, **`recover`**, or **`expunge`**. Minimal man pages should be written documenting these commands. (1/2 day)

Document DISCUSS_EDITOR variable

When the tty client edits a new transaction, it looks at the DISCUSS_EDITOR environment variable. This should be documented in the discuss man page, or the Discuss User Guides. (1/8 day)

Features that should be left to other projects

Mail transaction in tty interface

Since Discuss is often used to archive mailing lists, it would be convenient for users to be able to reply to the mailing list from within the Discuss tty client. This would allow the usual mailing features of including the original message, replying to all recipients, and adding the proper `In-reply-to` field to get proper chaining in the Discuss meeting.

This feature involves building the functionality of a mail sending program into Discuss. Discuss users use a variety of mail readers, and the Discuss interface will be different than their usual mail sending commands. The Discuss client would also need a way of knowing the mailing list address associated with the meeting, to set up the initial `To:` field. (2 days)

Flagged message in mail

When mailing to a list archived by a Discuss meeting, it would be convenient to be able to set the Discuss flags in the message header. A new message header, such as `Discuss-flags:`, could be used to list the named flags that should be turned on when the transaction is entered. (1/2 day)

Included `.meetings` file for security, etc

Currently, all the user's meetings are stored in a single `.meetings` file. Some users may wish to separate public and private meetings into two `.meetings` files, and set the file protections accordingly. Being able to include a `.meetings` file in another one, or maintain two files, `.meetings` and `.meetings.private` would make it easier for users to attend private meetings without restricting access to their entire `.meetings` file. (1 day)

Setting `.meetings` file

Some users may wish to maintain multiple `.meetings` files for different sets of meetings, and be able to switch the file from within the Discuss client. The quiet meeting requirement, listed above, may reduce the demand for this feature. (1 day)

Make `lsm -user <name>` or `lsm -public` work

The original design of Discuss allowed the `list_meetings` command to refer to list other users' lists of meetings, or a public list of meetings. In the Athena environment of attachable user lockers, this no longer works the same as time-sharing systems. The Discuss client could call `attach` or use AFS semantics to locate other users' `.meetings` files, and make these commands work again. (2 days)

No core dumps on bad acl file

Discuss acl files are text files that can be edited if necessary. The first line is a count of entries, following by the entries themselves. The `cacl` file is in the same format, controlling who can create meetings on a Discuss server, and must be edited by hand. The Discuss server will core dump if the acl file is bad, such as having an improper count.

The Discuss server should be more tolerant of errors. (1/2 day)

Manually resetting connections in clients

The Discuss client maintains connections to the different Discuss servers with which it is communicating. If one of these servers goes down, the user will have to quit the Discuss client to reestablish a connection. The Discuss client should have a way of manually resetting the connection. (1 - 2 days)

Printing out remaining number of transactions in print new

When printing new, the Discuss client should display the number of transactions left. This feature will not be easy to implement if deleted transactions have to be taken into account; the client will have to count up non-deleted transactions in the range. (3 days)

Built-in pager to back up

The Discuss client uses the `more` command as the default Discuss pager, by feeding the transaction text as standard input. `more` does not allow users to back up in this mode. Although the alternate pager `less` allows users to back up, most users do not use `less`.

The Discuss client should allow users to back up when reading transactions, either by using its own pager, or writing transactions to a file before running `more`. Adding a pager to Discuss would require a large amount of development time. Writing transactions to a file before running `more` will cause a significant delay when reading transactions. (5 days)

Command history

The tty Discuss client should maintain a command history so that the user can go back and redo previous commands. This would make it easier to enter repetitive or similar commands.

Discuss uses the ss library to implement the command-line interface. The ss library would have to be changed to use a history mechanism, and interpret the escape sequences to access the command history. This is a major addition to the ss library. (1/2 day)

Support discuss meetings in WAIS

The Wide Area Information Server (WAIS) Project is an experiment in automating the search and retrieval of many types of electronic information over wide area networks. It provides a standard protocol of obtaining information based on keywords. It would be nice to allow users to access the contents of Discuss meetings using WAIS. This would expand the distribution of text stored in Discuss meetings, and allow the Information Retrieval (IR) techniques in WAIS to be used to search Discuss meetings.

Supporting WAIS would involve major effort; this is an entire new protocol, based on text indexing. To do it properly, a Discuss meeting would need to maintain an text index on its contents, so that WAIS users could quickly find relevant transactions. Although the WAIS distribution contains code to perform this function on text files, changing it to understand Discuss meetings would not be easy. (10 - 15 days)

Redesign `.meetings` for seen/unseen transactions

The **`.meetings`** file currently keeps track of the highest transaction seen in a meeting. If a user skips a number of transactions, those transactions are essentially marked as "seen".

Discuss should keep track of whether individual transactions are seen or not. This would involve changing the **`.meetings`** file format to contain an arbitrary amount of information regarding the state of all transactions. Although tracking ranges of transactions should cut down the amount of information, Discuss must allow for the general case. This is a major change to the **`.meetings`** file, and requires changing a large amount of code. (5 - 7 days)

Subject/author based kills

In discussion groups, certain topics or authors start to generate relatively less useful information. Usenet allows users to set up "kill files", which automatically avoids messages from certain people or on certain topics.

Discuss should have a similar capability, controlled by some user kill file. The Discuss user interface would need a way to interpret the user kill file, and apply it to the set of transactions to print. The user would need some way to override the kill settings when displaying given transactions.

Since only a few Discuss meetings get transactions with low information content, this feature has limited usefulness relative to the cost of implementing it. (5 days)

Sets of transactions, either locally or globally

Discuss should allow users to keep track of a selected set of transactions, either locally to the user, or globally to all attendees of the meeting. The named flag feature seems to accomplish the latter for a limited number of sets.

Implementing this feature would either involve changing the `.meetings` file format, or having Discuss access other files for tracking Discuss transaction numbers. Either way would involve a lot of work, with unclear benefits. (3 days)

Check meetings in parallel

The Discuss client currently checks meetings one at a time for all servers. The check meetings operation would be faster if multiple servers were checked in parallel. This feature would involve adding the check meetings logic into the Discuss library, and use asynchronous calls to check when to receive the replies and make new requests. While this would bring speed improvements, it is not easy to implement. (5 days)

Expiring transactions

Discuss transactions are stored in a meeting until a user manually deletes them. For meetings that store recent mail messages, older messages should automatically expire and be deleted in some fashion. This could either be a function of the Discuss server, or a utility program that searches for transactions older than a given time and automatically deletes them. (2 days)

Expunge as a remote operation

To expunge a Discuss meeting, the administrator of the Discuss server currently runs a program on the Discuss server itself. It would be nice if a meeting chairman could expunge a meeting remotely.

The expunge operation copies the meeting, and requires an adequate amount of free space to hold the new copy of the meeting. The meeting chairman should have some way to find the amount of free space on the Discuss server to know if an expunge will work. Multiple chairmen could expunge meetings at the same time, causing the expunges to fail. These types issues should be resolved before remote expunging is enabled. (2 days)

Way of setting Zephyr-flag from a program

There is a per-meeting option to enable or disable Zephyr notifications for a Discuss meeting, although the only way to set this option is by manually editing the meeting file with `adb`. There should be a way to set this flag remotely, or have a user friendy utility that sets it. (1/2 day)

Speed up expunges

The expunge process is currently optimized for correctness rather than speed, causing expunges of large meetings to be a time-consuming process. With careful programming, the process could be sped up. (3 - 4 days)

Deal better with expunging a full partition

Expunging a meeting makes a copy of the meeting on the same partition as the original. If a Discuss server fills up a partition, freeing up space is a challenging procedure.

There should be an easier way for dealing with a full partition. Either the expunge process should use temporary space from a different file system, or provide a mechanism for temporarily moving some meetings to another file system. (1 day)

Ability to override acl's without editing manually

If the chairman of a meeting is unavailable, the only way of changing the access of a meeting is manually the acl file for the meeting. While this is not especially difficult, it is an annoyance, especially with the count of acl entries in the acl files. There should be a way for a Discuss administrator to override or set an acl without editing the acl file itself. (3/4 day)

Mac Discuss client

The Mac Discuss client should become a supported client, and the current bugs and user interface problems fixed up.

This task is too large to be considered part of this project. (15 days)

PC client

There should be a PC Discuss client.

This task is too large to be considered part of this project, although porting the tty interface over to DOS might be easier than having a graphical user interface. (60 days)

Statistics (counts) on searches

When specifying transactions based on flags or contents, there should be a way of just getting the count of transactions that match the criteria, rather than a listing of the transactions themselves. One way of implementing this is the ability to redirect a listing of transactions to a file, so that the word count `wc` can be used. (2 days)

Chains as trees

Discuss currently keeps track of transactions as linear lists of replies. If a discuss beings to fragment into a number of related topics, it would be nice if the transactions were organized into a tree.

This would complicate the user interface and methods of organizing transactions. Also, it seems difficult enough to get users to reply to transactions to form chains; it is not clear how many could deal with tree structures. (7 days)