# Grassroots: A System Providing a Uniform Framework for Communicating, Structuring, Sharing Information, and Organizing People

*Kenichi Kamiya, Martin Röscheisen, and Terry Winograd*

Computer Science Department
Stanford University, Stanford, CA 94305
http://pcd.stanford.edu

**Abstract**

People keep pieces of information in diverse collections such as folders, hotlists, e-mail inboxes, newsgroups, and mailing lists. These collections mediate various types of collaborations including communicating, structuring, sharing information, and organizing people. Grassroots is a system that provides a uniform framework to support people's collaborative activities mediated by collections of information. The system seamlessly integrates functionalities currently found in such disparate systems as e-mail, newsgroups, shared hotlists, hierarchical indexes, hypermail, etc. Grassroots co-exists with these systems in that its users benefit from the uniform image provided by Grassroots, but other people can continue using other mechanisms, and Grassroots leverages from them. The current Grassroots prototype is based on an http-proxy implementation, and can be used with any Web browser. In the context of the design of a next-generation version of the Web, Grassroots demonstrates the utility of a uniform notification infrastracture.

## 1. Introduction

People are currently using an array of disparate systems in order to keep up with new things happening, to file information they want to keep, and possibly to share them with others. Let us look at Tom, a user. Tom usually starts the day by seeing what has happened. To do this, he has to make a tour: First, he checks his e-mail (he looks through the messages which are queued up in his electronic mailbox). He reads a certain number (the rest remain in his mailbox folder), deleting some of the messages, sending replies to others, and filing the ones which he wants to keep into various mail folders in his (private) folder hierarchy. He shares some items with others by forwarding them. Tom next uses his newsgroup reader to read news much in the same way as he does with his e-mail. Then he uses a Web browser to check a number of hypermail archives for new messages in corresponding mailing lists (because he did not want to have all the mail in his e-mail box); there may of course be no new messages in some of these archives. Finally, he runs through a special list of Web pages to check whether they were updated;

for instance, this list includes pointers to pages about conference announcements which were labeled “Check this page regularly for the latest information.” Of course, he also keeps a hierarchical Web hotlist in addition to his e-mail and newsgroups folders.

What Grassroots does for such practice is to provide a uniform interface that is integrated at the activity level (and not only at the application level[1]). When using Grassroots, Tom would catch up with what happened by simply pressing one single button “goto notifier.” This would show him a structured description that summarizes all the events which he wanted to be notified about in one form or another. He selects here which event he wants to view. Note that the events are already structured (e.g., by topic or by chronology); this is not achieved through the use of filters, but instead by the way Grassroots makes structure visible to communication partners. For example, a student sending a brief question to his advisor posts his message directly into his advisor’s folder for student questions which can be answered quickly; the advisor typically sets up the notifier in a way which gives messages in this folder higher attention, and the notifier then signals arrival of such new messages accordingly. Generally, from the notifier summary, Tom can file the new items into their default position in an hierarchical archive structure by a simple approval; they can also be filed into any other place, of course, if specified.

Specifically, the archive structure is a hierarchy with “collectors” as nodes (like e-mail folders, but possibly shared); such collectors trigger appropriate events. For instance, when a message is inserted into such a node, this may trigger notification to other users.

Grassroots-external information such as an hypermail archive pages or the previously mentioned conference page can be declared to be collectors, and thus whenever links are added to them (manually or by virtue of a message having been sent to the corresponding mailing list), this is appropriately triggered through the Grassroots system. The use of Grassroots is transparent to non-Grassroots users. For instance, regular mailing list subscribers will not notice that a Grassroots user receives the same information as part of a uniform notification and communication mechanism; the Grassroots user also does not observe that a reply to a hypermail node really ends up appearing as a mailing list message to most readers.

Thus, Grassroots is not only a system for personal information management, but its ability for access-controlled sharing also enables the continuous construction of shared representations such as “organizational memory” (when used in organizations). Moreover, since Grassroots also makes visible the underlying network of information flow, it makes it easy for new members of a community to get attuned to the “association

---

[1] Bundling together a Web browser, a news reader, and a mail reader such as done in Netscape Navigator 2.0 is an example of what we mean by integration at the application-level. This is already useful, but we are interested here in a more uniform integration which goes beyond the historic artifact of such applications being distinct with boundaries which are cut across by actual usages.

forms" [Agre, 1995] of this community; this includes being able to quickly learn about where members of this community receive their information from, etc.

In this paper, we lay out the general framework of Grassroots and describe how Grassroots realizes the various components of this framework. We describe how Grassroots provides a uniform interface to usages and practices currently found in various communication and information management systems. Furthermore, we examine the user interface of the current Grassroots prototype, and describe preliminary observations stemming from its use.

# 2. Collection-Mediated Collaboration and The Conceptual Model of Grassroots

The activities alluded to in the introductory examples have one thing in common: they are mediated by collections of information such as folders, hotlists, e-mail inboxes, newsgroups, and mailing lists. These collaborative activities mediated by collections are what Grassroots aims to support. We group these activities under the heading "collection-mediated collaboration." Grassroots represents and supports collection-mediated collaboration through a simple model with only four classes of objects–collectors, notifiers, links, and articles–and a small number of generic operations. In this section, we describe the general framework of collection-mediated collaboration and introduce Grassroots' conceptual model for realizing the framework.

We find that the daily information activities which are mediated by collections of information can be understood by three major concepts: units for statically grouping collections of information ("collectors"), views for presenting "new" information to each person ("notifiers"), and modes by which people communicate by transferring information between collections. In the remainder of this section, we look at each of these in turn, first examining current practice in existing systems and then how these concepts are reflected in Grassroots.

## 2.1 Units for Organizing Information and People: "Collectors"

**Collectors in Current Practice**

The most important concept in understanding collection-mediated collaboration is the concept of containers that accommodate collections of information ("collectors"). The following are examples of commonly used collectors.

- *Information Categories/Topics:* People organize information by creating collectors that represent categories and by sorting and storing pieces of information into these collectors. Macintosh folders are examples of such collectors.
- *Groups for Collective Communication:* One might want to make a list of a certain group of people and give information to or receive information from them. Such

collectors consist of identifiers of the group's members. A mailing list is an example of collectors used for such collective communication.

- *Groups for Access Control:* When collectors are shared, they need to be access-controlled for various social reasons. A collector can be used to represent a group of people who are permitted to access some other collector.
- *Relationships among Collectors:* Generally, one would like to link collectors together in the form of a graph. To represent relationships among collectors, another type of collectors is used. This is illustrated by a hierarchy of collectors, in which an intermediate collector lists sub-collectors. Yahoo is an example of such a hierarchical information collection structure where each page can be thought of as a collector. The relationships among collectors that stand for human groups can be represented in the same way.

In other words, we define a collector as a unit for organizing information and groups.

### How Collectors Are Reflected in Grassroots

The Grassroots collector (g-collector in short) is one of four main object types of Grassroots. The collectors described above can be represented by g-collectors and manipulated with the same set of generic operations. For example, a specific g-collector can be simultaneously used as a mailing list and as an access control group. Moreover, anyone can construct a g-collector representing a group of people by simply pasting their addresses into the g-collector in the same simple fashion as one would insert a document into a hotlist.

In the model of Grassroots, g-collectors can contain *articles* (pieces of text such as e-mail messages or news articles), *links* (pointers to other g-collectors or to external objects such as Web pages), or *other g-collectors*. (Cf. Figure 1.) There are two sub-types of g-collectors: g-collectors for structuring information are *folders*; g-collectors for organizing people are *rosters*. Each g-collector has uniform address (a URL in the prototype).

In a roster, a link to one of a person's g-collectors represents the person. When a roster is used as a mailing list, mail is brought to the g-collectors linked from the roster since Grassroots' mail is not addressed to a person but addressed to one of a person's g-collectors. When a roster is used for controlling access to a g-collector X, for example, the owners of the g-collectors linked from the roster are authorized to access the g-collector X.

Under a g-collector, a user can create objects (links, articles and other g-collectors), change their attributes, cut or copy them, and paste them into another g-collector. (Cf. Figure 2)

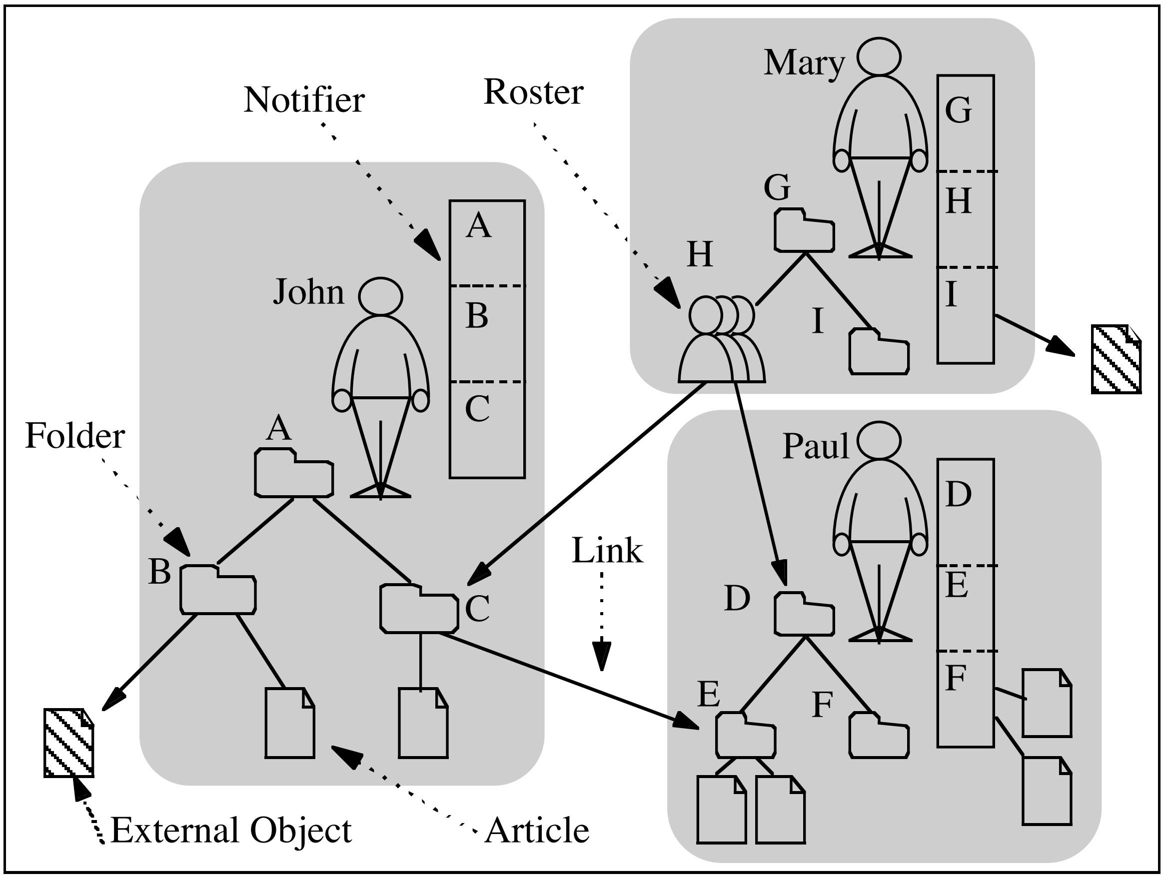



**Figure 1. Grassroots Objects.**

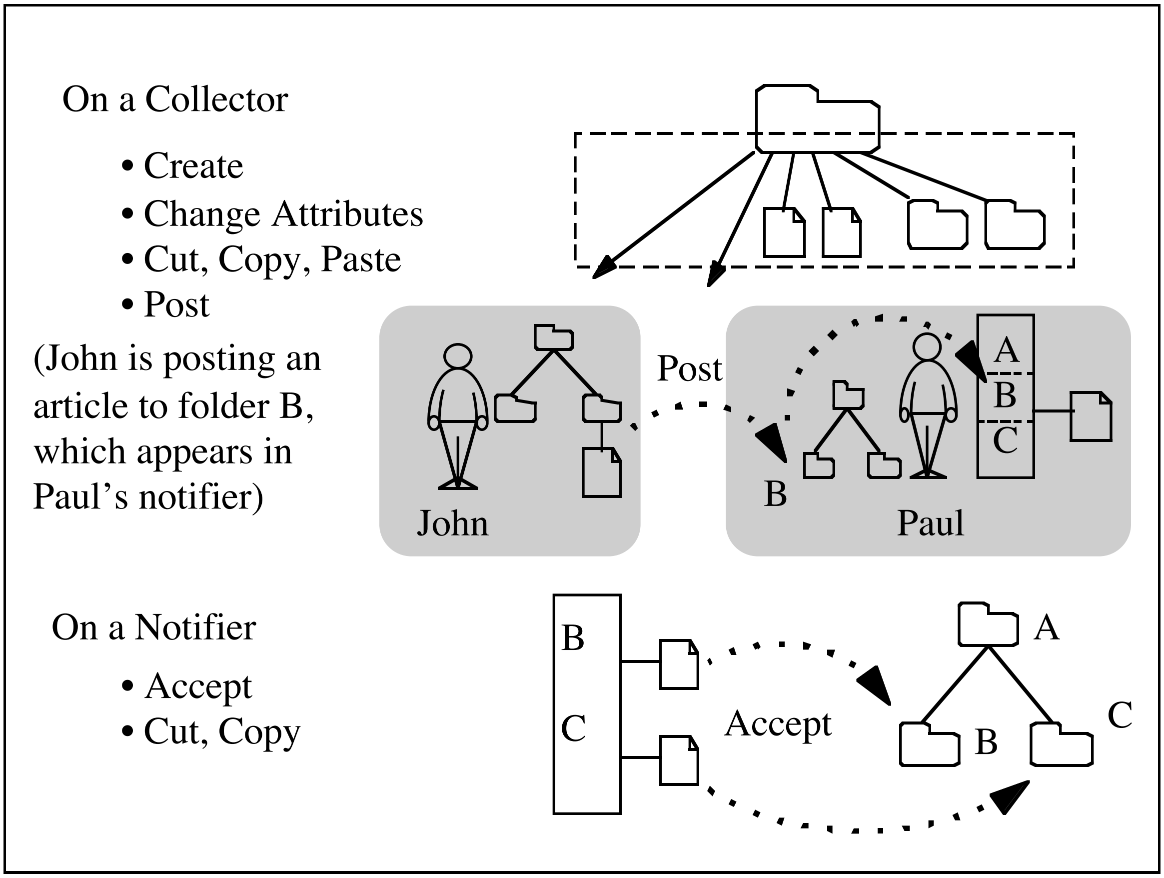



**Figure 2. Grassroots Operations.**

## 2.2 Views for Presenting New Information: "Notifiers"

**Notifiers in Current Practice**

When information is added into a collector manually by other people or automatically by a system, the owner of the collector might want to be notified of the addition. In other words, people like to see "what's new" in some collectors, especially in their own collectors. Usually there are views that notify them of such new events by listing or highlighting new pieces of information. We call them "notifiers". E-mail readers and news readers are examples. Operations on the notifier often include such actions as marking items as "done" and filing them into some collectors, keeping items in the notifier for later renotification, or deleting items (and not being renotified about identical events in the future). A problem found in current practice is that disparate notifiers present a person information arrived by different modes such as mail and news. It is desirable for these notifiers to be unified into one notifier.

**How Notifiers Are Reflected in Grassroots**

In Grassroots, each user has one notifier. A user's notifier presents the information items brought to the user's g-collectors without involving the user directly. A notifier is automatically divided into sections corresponding to the structure of the g-collectors; then each section lists the items that arrived at a corresponding g-collector.

In addition to the operation described in the previous section, a user can *post* items on the clipboard (which has been cut or copied) to another user's folder with an operation similar to pasting the items to the folder. In Figure 1, if John posts an article to folder E, the article does not appear in folder E. Instead, the article appears in section E of Paul's notifier. The recipient can choose some of the objects in a notifier, and *accept* them; they are then moved and inserted persistently into the corresponding g-collector (Figure 2). A user's notifier can be read only by the user.

A user can also post an article (or a link) to a roster. This is equivalent to posting the object to all g-collectors linked from the roster or its sub-rosters. This way, a roster can be used as a mailing list.

Generally, a collector can be conceptualized as having a store for a collection of persistent items, a store for a collection of newly added items, and three ports, one for pasting things into it, one for posting things into it and another for items coming out of it. As shown in Figure 3, *post* is an operation to insert an item into a folder with notification to the folder's owner while *paste* is an operation to do so without such notification.

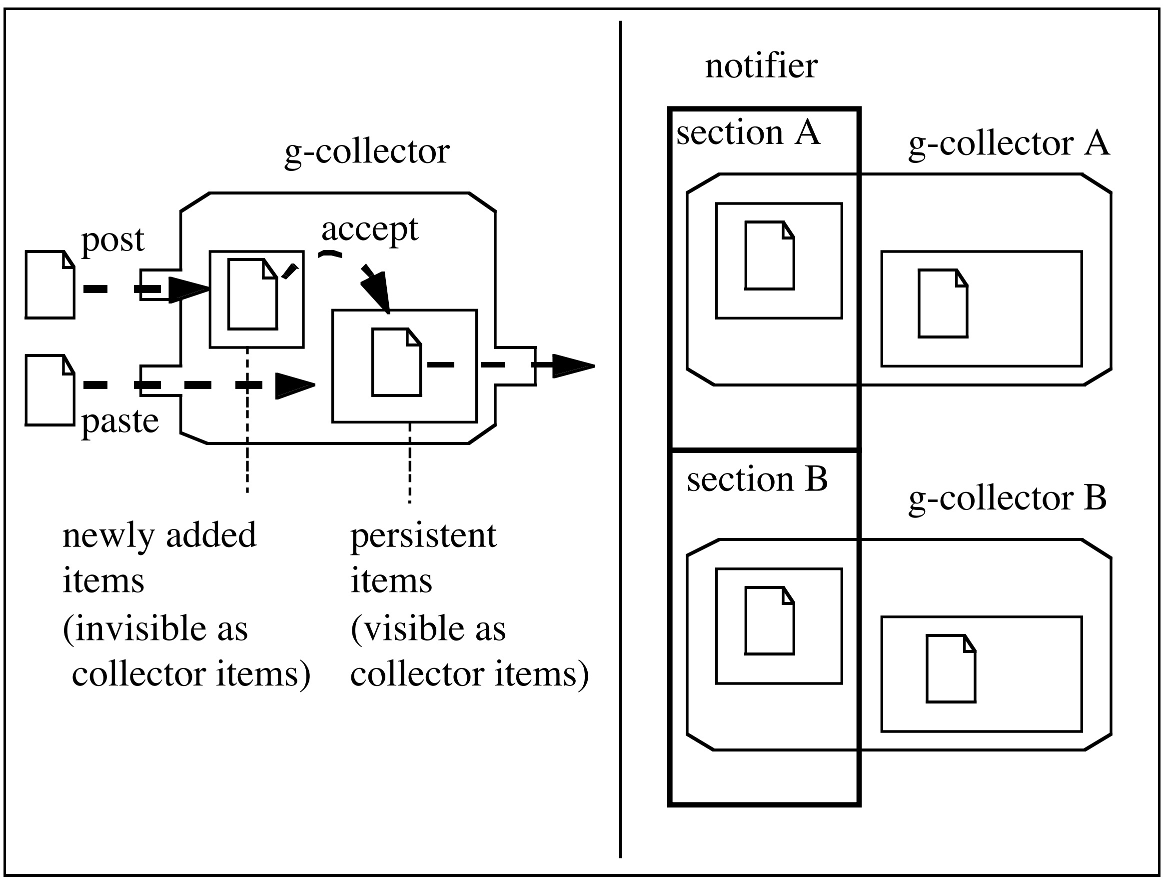



**Figure 3: G-collectors and a Notifier.**

## 2.3 Information Transfer between Collectors

When observing the way people collaborate with the help of collections of information, we find that they communicate with each other by transferring information between their collectors. For example, when people send a letter, the letter is transferred from their outboxes to the recipients' inboxes. When people subscribe to electronic newsgroups, articles are transferred from the newsgroups to their news inboxes, which conceptually exist at the recipients' side. In this section, we analyze how information items travel between people's collectors.

### 2.3.1 Transfer Modes: Information Pull or Push, Continuous or Ad hoc

**Transfer Modes in Current Practice**

Transfer of information between collectors can be characterized by its regularity, by who is initiating it, and by how recipients are notified of it.

One dimension to look at information transfer is its regularity. We consider a transfer to be in *continuous* mode if all the items incoming to the source collector are transferred to the destination collector, as is the case with newsgroup subscription. In the complementary mode, the transfer between a source and a destination is decided on a case by case basis for each item. We call this the *ad hoc* mode. For example, each e-mail has its own combination of the sender's outbox as a source and the recipient's inbox as a destination.

A continuous transfer of information items can be initiated by the sender (the administrator of the source collector) or by the recipient (the administrator of the destination collector). We call these modes the *pull* or the *push* mode of information transfer, respectively.

Information transfer can be performed with or without notification to the recipient. *With notification* means that items newly arrived at the destination collector are kept separately or marked distinctively so that someone in charge of the destination collector can distinguish them from older items. In other words, new items appear on the notifier of the administrator of the destination collector. *Without notification* means that new items are simply inserted into the destination collector.

In summary, people transfer information in one of six modes: *ad hoc*, *continuous pull*, and *continuous push* either with or without notification. Examples are given in the following table.

| | With Notification | Without Notification |
|---|---|---|
| **Ad hoc** | Messages in one's outbox are transferred to recipients' inboxes. | Items are copied from one hotlist and pasted into another hotlist. |
| **Continuous Pull** | Articles in a newsgroup are transferred to the subscribers. | Articles in a newsgroup are transferred to a hypernews page. |
| **Continuous Push** | Messages arrived at an e-mail account are transferred to another account by a ".forward" file. | Messages arrived at a mailing list are transferred to a hypermail page. |

**Table 1. Six Modes of Information Transfer: Examples.**

### How Transfer Modes Are Reflected in Grassroots

In Grassroots, these six modes can be realized by generic operations including setting and unsetting attributes of objects (cf. Table 2). The operations corresponding to ad hoc modes have been introduced in the previous section.

| | With Notification | Without Notification |
|---|---|---|
| **Ad hoc** | *copy &post* (operations) | *copy & past* (operations) |
| **Continuous Pull** | *subscription* (an attribute of the link) | *noticeless subscription* (an attribute of the link) |
| **Continuous Push** | *forward* (an attribute of the link) | *noticeless forward* (an attribute of the link) |

**Table 2. Six Modes of Information Transfer: Grassroots Elements.**

Every Grassroots object has a number of attributes, such as the name, the date of creation, etc. *Subscription, forward, noticeless subscription,* and *noticeless forward* are attributes of the links. These attributes can be set and unset. If they are set, information flows continuously between g-collectors at the both end of the link. Subscription enables the continuous pull mode while forward enables the continuous push mode.

Consider a folder C, owned by a user John, contains a link pointing to Paul's folder E (as in Figure 1) and the link's *subscription* attribute is set. Whenever a new item (a link or an article) is brought into folder E, a copy of the new item will automatically appear in John's notifier in section C. If the *noticeless subscription* attribute of the link is set, a copy of the new item appears in folder C.

Similarly, if the same link's *forward* attribute is set, then whenever John puts an object into folder C, its copy will automatically appear in Paul's notifier in section E. If the *noticeless forward* attribute of the link is set, a copy of the new item appears in folder E as long as John is authorized to write into folder E.

A roster can be used to subscribe to or forward to a set of g-collectors at once. A subscription link (a link with its subscription attribute set) from a folder to a roster is equivalent to subscription links from the folder to all the g-collectors linked with the roster or its sub-rosters. The same applies to forwarding.

### 2.3.2 Inflow and Outflow of Collectors

**Inflow and Outflow in Current Practice**

People transfer information to and from collectors which they use for collaboration. We call the transfer to and from a collector the collector's inflow and outflow. Reading or displaying information items in a collector is not considered outflow from the collector. Instead, copying or moving one collector's items to another durable collector is considered the outflow from the former collector and inflow to the latter collector. Both inflow and outflow are performed in some of the four modes of transfer described above.

In current practice, people often use collectors with only one mode of inflow and one mode of outflow. Since there are six modes of transfer, we can unfold this into a fine-grained grid of 36 types of collectors. Since the difference between *with notification* mode and *without notification* mode is trivial for outflow, the number of types is 18 instead of 36. This is shown in Table 3 with examples for some of the types.

| Inflow / Outflow | ad hoc | | continuous pull | | continuous push | |
|---|---|---|---|---|---|---|
| | without notification | with notification | with-out N. | with N. | with-out N. | with notification |
| ad hoc | a shared hotlist<br>a private hotlist | a Yahoo page<br>a mail inbox | a hyper-news page | a news inbox | a hyper-mail page | a mail inbox with its address on a mailing list |
| continu-ous pull | a newsgroup | a moderated newsgroup | | | | |
| continu-ous push | the set of messages sent to a mailing list | the set of messages sent to a moderated mailing list | | | | |

**Table 3. Eighteen Types of Collectors: Examples.**

For example, Table 3 describes a hotlist (shared or private) as a collector whose inflow is *ad hoc without notification* mode and outflow is *ad hoc* mode. This is because the items are put into it on a case-by-case basis without explicit notification to anyone, and since the items in it can be copied to somewhere else on a case-by-case basis. In contrast, items are transferred to a Yahoo page or a mail inbox with explicit notification to someone in charge of those collectors. Newly arrived items are kept separately for evaluation by an editor or marked distinctively so that the recipient can recognize them. Therefore, inflow to them is *ad hoc with notification* mode.

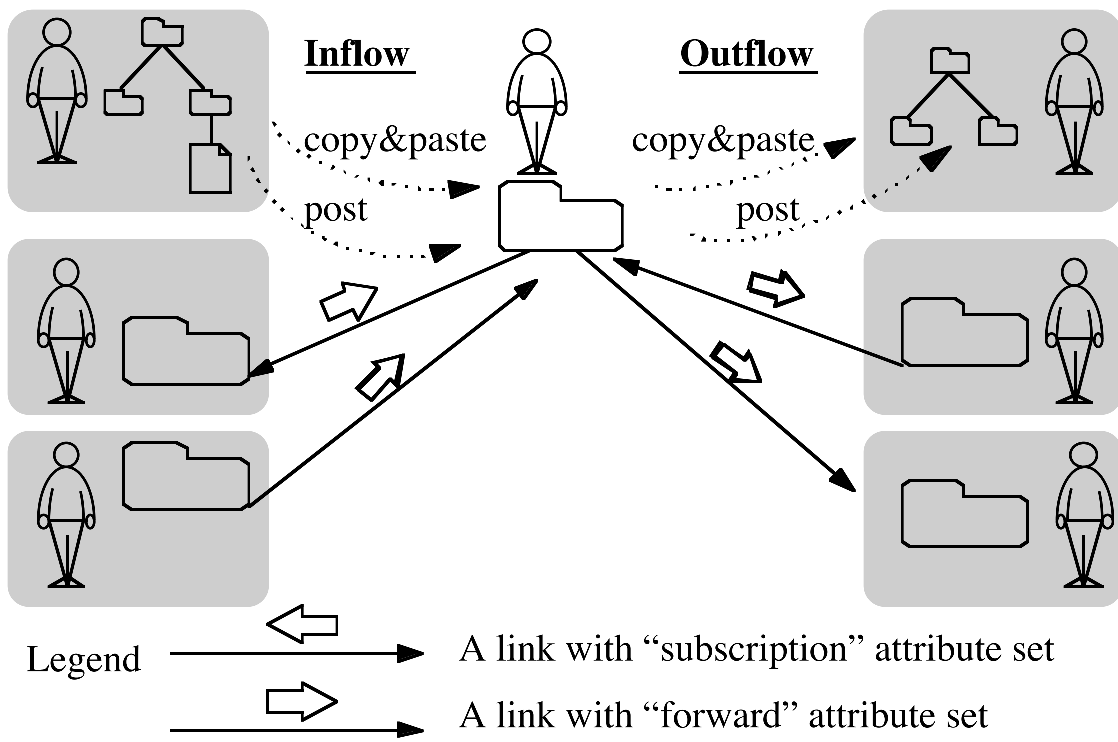
When a person subscribes to a newsgroup, all the items incoming to the newsgroup are transferred to the news inbox, which conceptually exist at the subscriber's side. This is similar to the situation involving a mailing list, where all the items arrived at the mailing list are transferred to the mail inboxes whose addresses are on the list. One of the differences between them is who controls the transfer. Newsgroup subscription is decided by the subscribers, not by someone in charge of the newsgroup. For this reason, the outflow from a newsgroup and the inflow to a news inbox are *continuous pull* mode. In contrast, the administrator of the mailing list controls whether to have an address on the list or not. Therefore, *continuous push* mode applies to the inflow to a mail inbox whose address is on a mailing list and outflow from the conceptual set of messages arrived at a mailing list. A hypermail page receives mails in the same manner as a mail inbox whose address is on a mailing list, while a hypernews page receives articles in the same way as a news inbox.

Usually a newsgroup receives articles spontaneously from various people. A mailing list receives mails in the same fashion. Therefore, their inflows are both *ad hoc* mode. When they are moderated, their inflows are *ad hoc with notification*, since the newly arrived items are not inserted directly but kept separately on the moderator's notifier.

The outflows from the collectors in the first row of Table 3 are all ad hoc since, usually, items incoming to these collectors are transferred to somewhere else spontaneously or are not transferred to anywhere at all.

**How Inflow and Outflow Are Reflected in Grassroots**

Grassroots provides elementary objects and operations that correspond to six modes of transfer. Any mode can be used as inflow to or outflow from a g-collector as depicted in Figure 4. By combining these elements, a g-collector in Grassroots can be used as any of the eighteen types described above. A g-collector can be flexibly transformed from one type to another by adding or removing appropriate elements. In addition, a g-collector is allowed to have any combination of four modes as inflow or outflow (instead of having only one inflow mode and one outflow mode).




(Thin solid arrows depict links. Thick open arrows depict the direction of the flow of information items.)

**Figure 4. Inflow and Outflow: Any mode can be used.**

## 2.4 Controlling Flow with Access Control

**Access Control in Current Practice**

In current practice, access control to collectors is sometimes not very flexible or might even not be available, such as in newsgroups. Shared hotlists on Web servers can be access controlled, but a user has to register a password for each server. As for shared filing, in many systems operations for manipulating the access control list and that for manipulating data are not uniform. As a result, for people other than system administrators, it is not very easy to set up a group for access control or to control access using a group.

**How Access Control Is Reflected in Grassroots**

Whether an item can flow into a g-collector or whether it can flow out of it is uniformly constrained by an authorization mechanism based on access control policies.

Users have an identity representation with their g-collectors; they are expected to be authenticated on a per-user basis. Access control policies can be articulated with respect to groups of people. Such access-control groups are just the same kinds of human-organizational g-collectors which have also been used to represent mailing lists, etc. In other words, it is as straight-forward for an end-user to maintain an access-control group as it is to maintain a hotlist of specific documents.

In the model of Grassroots, access is controlled by three attributes of g-collectors, *read protection*, *write protection*, and *post protection* and three attributes of links, *read authorization*, *write authorization*, and *post authorization*.

For example, in Figure 2, if folder C's *read protection* attribute is set, no user except for John can read the contents of folder C. However, if *read authorization* attribute of the link from folder C to folder E is set, Paul, can read the contents of folder C. Similarly, if *write authorization* or *post authorization* attribute of the link is set, Paul can change the contents of folder C or post articles to folder C.

A roster can be used to authorize a group of people at once. A read-authorization link (a link with *read authorization* attribute set) from a g-collector to a roster is equivalent to read-authorization links from the g-collector to all the g-collectors linked with the roster or its sub-rosters. Then, owners of all g-collectors linked with the roster or its sub-rosters are authorized to read the g-collector. The same applies to other authorization rights.

# 3. Usages of Grassroots

## 3.1 Using Grassroots for Usages Found in Existing Systems

Grassroots provides a uniform interface to functionalities provided by existing systems such as e-mail, newsgroups, or any of the other examples which fit into the table of 18 types of collectors in the previous section. Often, Grassroots is not only able to cover functionalities of existing systems, but also offers advantages over them.

### 3.1.1 Example: E-mail

A Grassroots folder can be used in the way an e-mailbox is conventionally used. People can post articles to a folder, and its owner can be notified of them through his or her notifier. The difference is that, unlike in conventional e-mail, Grassroots provides each user with multiple mailboxes (the folders), organized in a hierarchy that is visible to senders.

With many addressable collectors as mailboxes, Grassroots avoids the need for "intelligent filters," etc. at the receiver's side because the incoming information items are not indiscriminately mashed together into one stream in the first place. That is, under a conventional e-mail system, since a user generally has only one mailbox, all the messages addressed to her will flow into it. She then has to read and categorize the mail for processing or storage (possibly with the aid of filters, etc.).

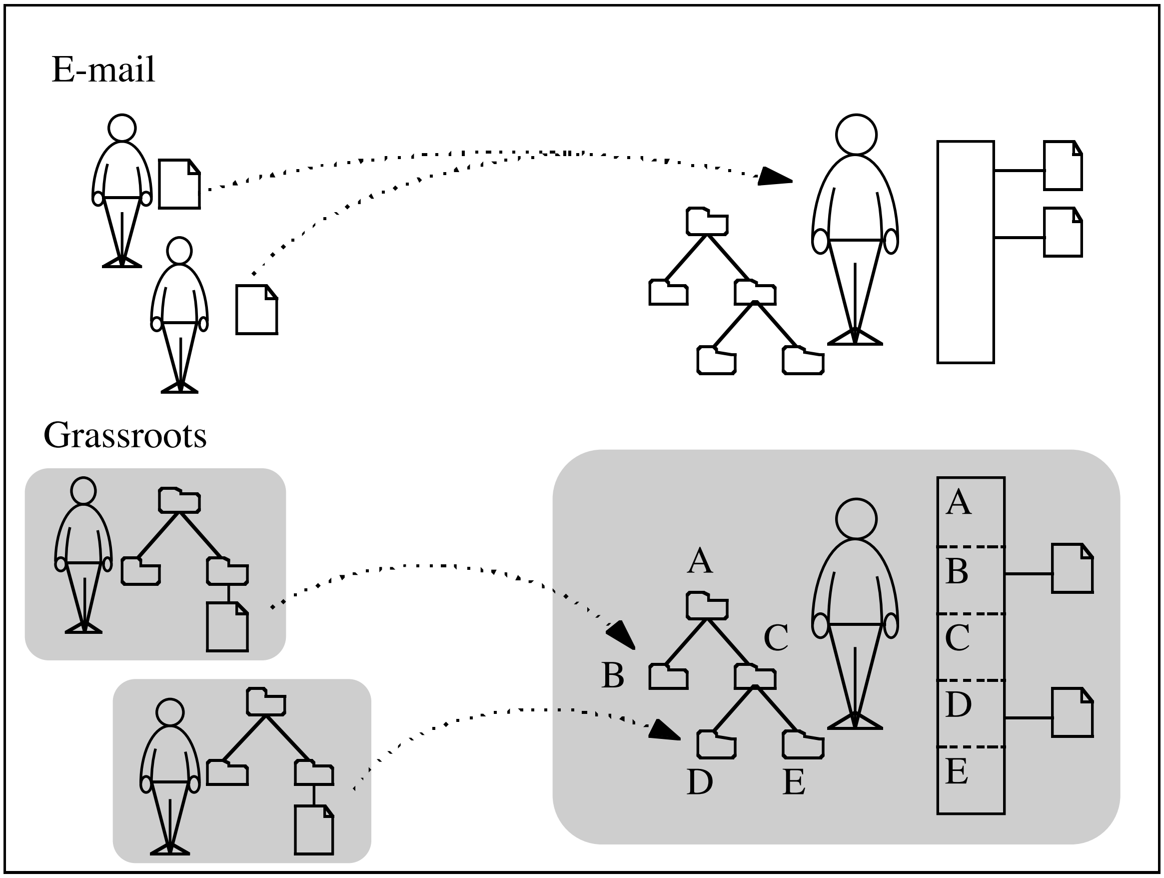



**Figure 5. Grassroots and E-mail.**

Using Grassroots, both senders and recipients can choose the destination folder to which items will be posted. A recipient can choose the destination by choosing one of his folders and giving its address to potential senders or to some mailing lists. A sender can choose the destination by browsing the recipient's folder hierarchy and choosing the folder that looks most appropriate to post objects to. The sender can also keep a link to that folder and post objects onto that link. When a Grassroots user opens his notifier, all objects are already sorted according to the organization of his collectors.

Note that often this is advantageous for both the sender and the recipient. In effect, Grassroots enables here the continuous development of a shared language between its cooperative users. Recall the introductory example of a student sending a message to his advisor. As a part of the shared language, the advisor's folder hierarchy might contain a folder for requesting brief feedback about simple questions. If the student does not abuse this folder, then the advisor will assign it a high notification priority, and both parties profit from having successfully coordinated themselves about a certain practice (the student receives quick feedback and is not stalled; the advisor can rest assured that his notifier will distinguish what can be answered quickly from other potentially time-consuming requests).

In Grassroots, if a user replies to a notifier item, the reply will be posted to the folder where the item came from. Subsequent replies by both parties go back and forth between the two folders. This allows a user to encapsulate a conversation and its records.

### 3.1.2 Example: Newsgroups

A folder of Grassroots can take on the role of a conventional newsgroup. This would be the case if people subscribe to a folder and post articles to it. By default, such a collector would be a moderated newsgroup since all posted items first appear on the folder owner's notifier. Only the accepted ones then appear in the folder. To have unmoderated discussion, it is possible to set the "accept all" attribute of a folder.

The Grassroots way of having newsgroups has advantages pertaining to ease of creation, access control, enabled competition, and as-needed-only replication.

*Light-weight, Decentralized Creation:* Note that creating a Grassroots folder by clicking on a button is much easier than running through the process of lobbying with your local system administrator (to create a new UseNet newsgroup) and with 100,000 other system administrators (to carry this new newsgroup at their site). Indeed, a user does not even have to worry about whether a certain topic is worth a wide distribution since creating a folder costs no more than creating a new Web page. This ease of setup enables and encourages the creation of "micro-newsgroups," that is newsgroups of a finer granularity in terms of topic or time of coverage. While some of such micro-newsgroups can be just for a short-term project, others might live long and grow to attract a large audience. Of course this advantage is made possible by the new network environment (Internet).

*Access Control:* The amount of replication inherent in the current newsgroup mechanism makes it intrinsically hard to enforce any kind of access control. Since new messages are forwarded from one news server to the next until all are reached, access control would have to be enforced at any of these sites. In other words, the distribution mechanism creates a vast trusted computing base—not the kind of assumption one would make to arrive at a secure system. The access control mechanism of Grassroots quite naturally extends access restrictions to its (micro-)newsgroups usage. It is then easy to set up a Grassroots newsgroup with restricted membership.

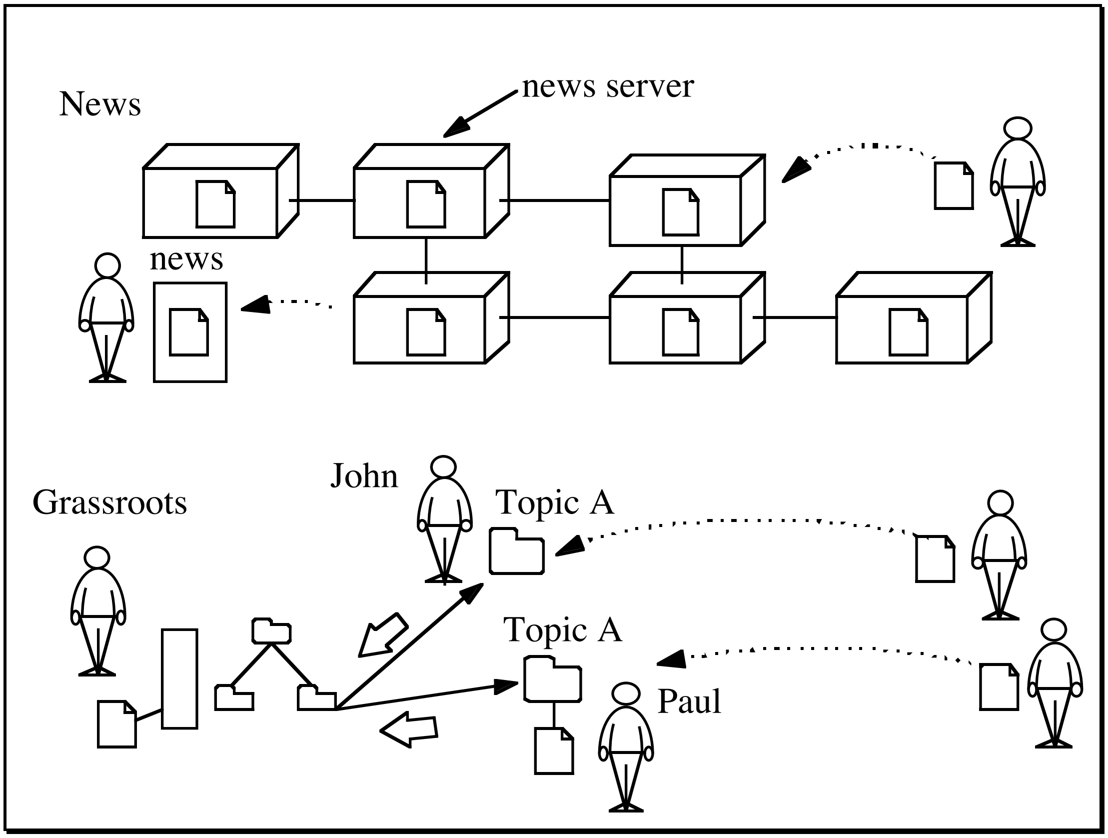



**Figure 6. Grassroots and Newsgroups.**

*Replication On an As-Needed Basis Only:* If we look at the way UseNet replicates news all over the place, we see that this is a good trade-off for the case where a particular item is read everywhere. However, for all other cases, when certain items are not read at certain sites, they are unnecessarily replicated there. Note that this is basically only a legacy from the historical development of newsgroups where, in the past, temporarily connected computers were dialing up each other at synchronized times and transferring bulletin board messages in a store-and-forward way (in the FIDOnet-like style which is still the primary mode in most developing countries). However, these days, we generally can rely on permanently connected Internet routers, and servers can be reached at any time. In this setting, at least one of the original motivation of message store-and-forward is lost, and it is then advantageous (in terms of distribution efficiency, access control, etc.) to choose the direct access model which the Web is now using (again enabled by a critical mass of machines being permanently on-line, of course). This is the way the Web-based Grassroots does it. Grassroots replicates news to a subscriber's storage only. If a news item is a link to a document (quite common in Grassroots), only the document's address is replicated to the subscriber's storage. The document itself remains in its original server, replicated only to the browser's memory when the link is selected. Note that if people do want to have replication for various reasons, this can be done in Grassroots with the subscription mechanism, by which items can be replicated from one "news server" to the next. Effectively, this realizes NNTP news distribution on a site and folder specific basis.

*Competition:* Finally, the ability to set up newsgroup in a decentralized way allows us to have more than one folder with the same topic; these can then compete for popularity. For example, if one newsgroup about a certain topic contains a lot of noise such as low-quality items or the same item repeated multiple times, people can move over to another newsgroup about the same topic (if there is any such other group; if there isn't, there would be reason to set one up).

What we are losing at a first sight when moving to Grassroots newsgroups is the ability to easily find relevant newsgroups. While UseNet offers a list of all newsgroups, Grassroots does not have such a list, nor would it seem a good idea to build into Grassroots the central maintenance of such a presumably quite large list for the diversity of newsgroups which might arise. This is a disadvantage at first. However, we can easily envision the creation of Yahoo-like services for helping people to find the newsgroups they are looking for. Note in particular that anyone can create in Grassroots a list of (links to) suggested newsgroups in his or her folders and make it available to other people (in one of the typical Grassroots usages as a Yahoo structure operated not as a service but as a continuously cooperatively maintained structure[2]). Such lists can be hierarchical, as in Yahoo; and competition among them is possible in a uniform way.

### 3.1.3 Usages from Other Systems

Each of the sixteen types of collectors has only one mode of inflow and one mode of outflow. However, a Grassroots collector allows any combination of the four modes as inflow or as outflow. For example, the folder depicted in Figure 7 forwards its new objects to primary members but other people can also get them by subscribing to it. This is a combination of a mailing list, its archive, and a newsgroup.

[2] We certainly don't want to be interpreted as intending to supplant Yahoo(TM) or anything like this. Indeed, pages of this highly useful service are quite usefully used as Grassroots collectors. Grassroots provides more a framework in which interoperability for Yahoo-type services can be defined. Also, there is no reason why certain Grassroots collectors couldn't be operated professionally.

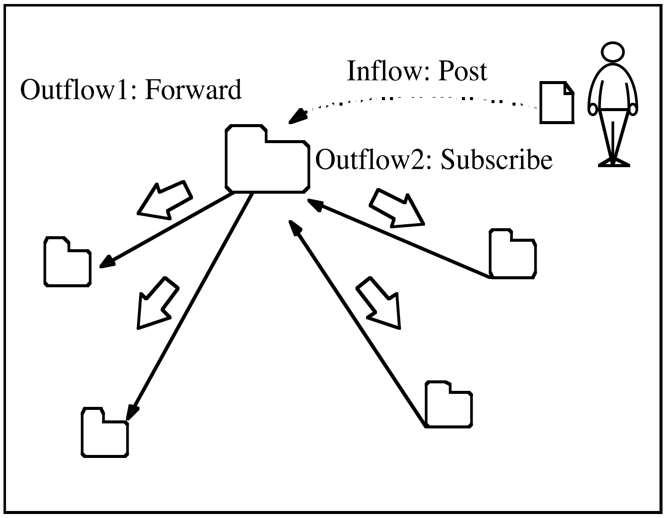



**Figure 7. Collectors with Different Modes of Outflow: Example.**

## 3.2 Continuous Cooperative Information Gathering and Routing

Grassroots realizes continuous cooperative information gathering and routing. It implements a form of social filtering with an articulated structure that is in contrast to experimental approaches based on comparing feedback statistics.

### Information Gathering

Using the subscription mechanism, Grassroots users can create a network for information gathering. When people are gathering information according to each person's interests, everyone profits from this by getting the latest findings from the other people who share the same interest. Grassroots enables such cooperation by letting all users subscribe to each other's folders (unless access restricted); in the larger context, this defines an information gathering network.

Such a network in effect implements a form of *social filtering*. Recall that even under the same topic different people gather different kinds of information into their collectors. A user can choose which of these collectors best match her interest and subscribe to them. Such a choice essentially filters the information flow. As a result of a subscription, the latest items of other people's collectors will be brought to the subscriber's notifier, where the user can then choose some of the items and accept them into the corresponding collector or disregard others. This choice does further filtering. In this way, subscribers can populate their own collectors and let others subscribe to them. Contrast this with UseNet newsgroups where users form their own combination of subscriptions, but no one else can see or take advantage of this combination.

In addition, Grassroots can make visible who gets information from whom. By following subscription links, one can find people who are working on or interested in a certain topic. Of course the folders that contain the links can be access controlled if needed.

Also, it is our experience with UseNet newsgroups that one can easily miss a relevant newsgroup because one doesn't notice since there is no practical way for people to compare their subscription lists. Learning to consult better information sources is thus made hard. In Grassroots, the use as a *referral network* is explicit. A user unfamiliar with one topic area (such as an employee entering a new company division, or an individual developing a new interest) can take advantage of the referral network. Browsing the network also makes clear which collectors are popular.

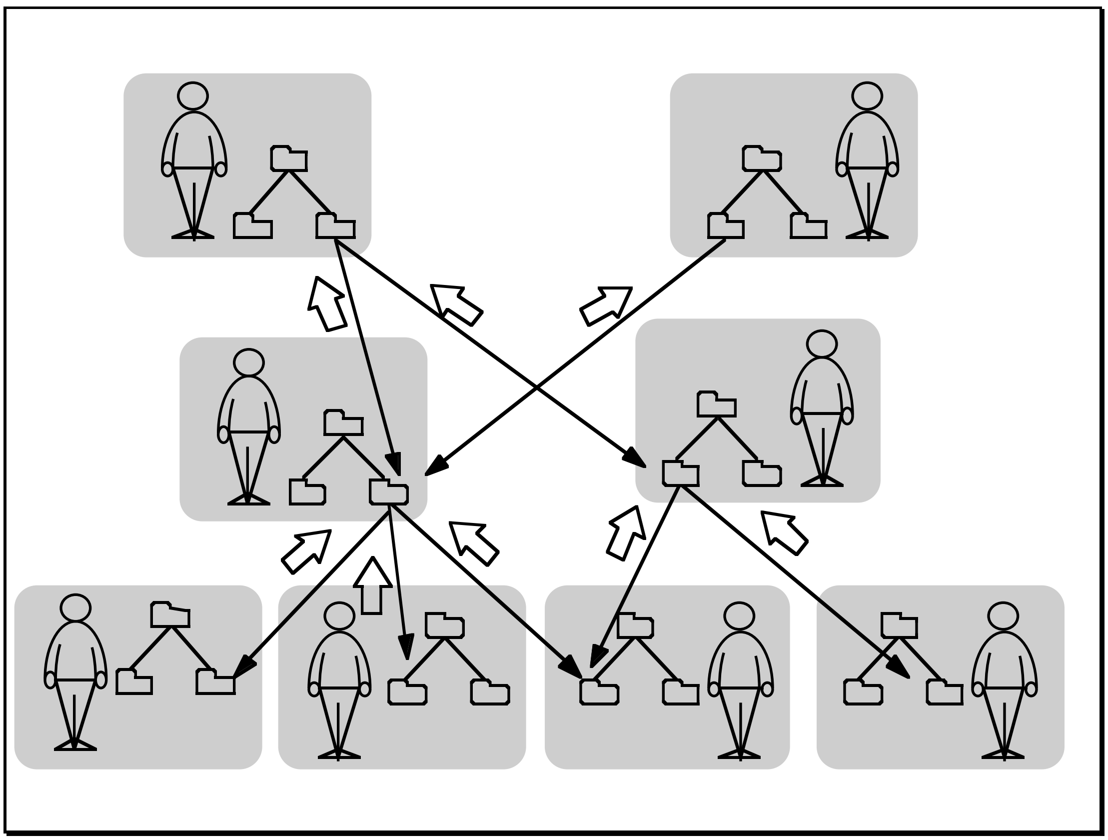


**Figure 8. Information Gathering by Chaining Subscriptions.**

**Information Routing**

The "forward" mechanism can be used as a tool to program a network for information routing. By chaining multiple forwards, one can create an "assembly line" for information objects. Each user on the route can contribute in adjusting a route according to a changing situation.

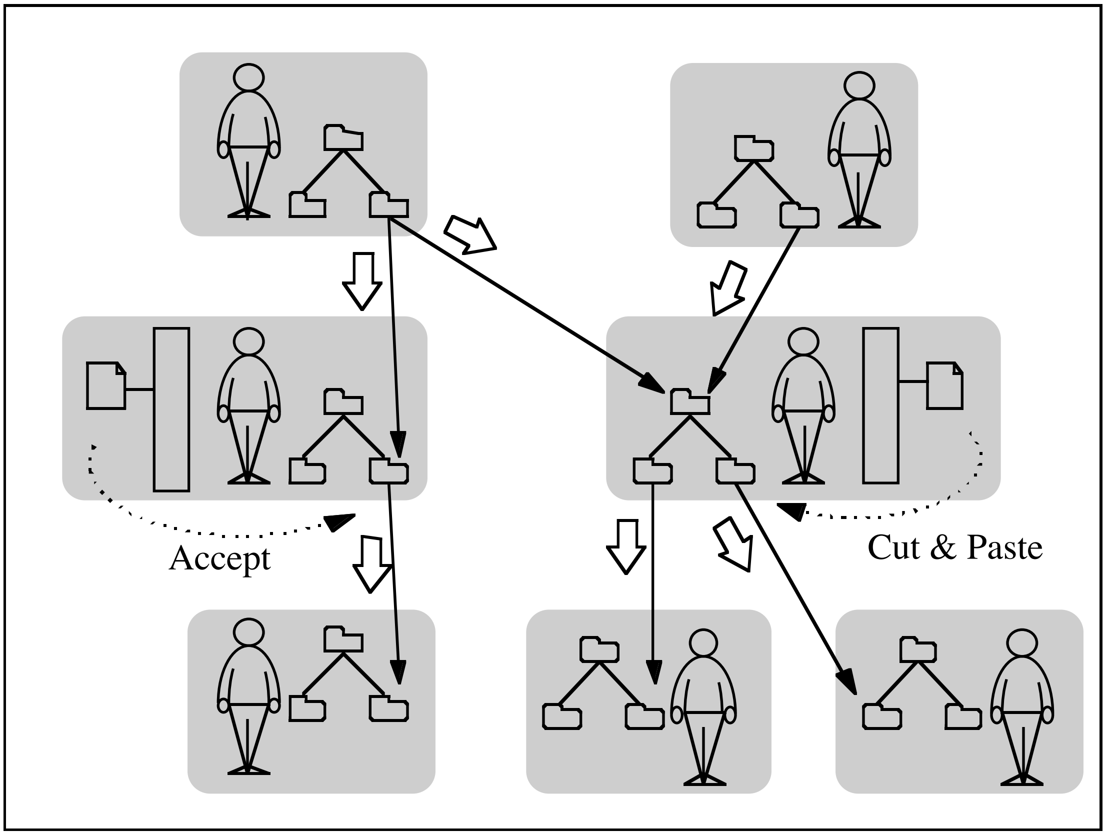



**Figure 9. Information Routing**

The way of chaining forwards can be direct or indirect. If two forward links are chained directly at one collector, then as soon as an item on the notifier is accepted, it will be forwarded to the next collector. If the object is a link, the notifier lets a user open the link location and process the source document before accepting the link object. In the case of indirect chaining, the user has to move the object to a collector which forwards it to the next user. In particular, it is useful that a route can have branches.

**Comparison to Statistical Systems for Social Filtering**

Social filtering in Grassroots contrasts with other social filtering systems such as Tapestry [Goldberg, 1992], GroupLens [Resnick, 1994], etc. (for newsgroups filtering) or any of the “agent” type systems (for instance, Ringo for music recommendations, etc. [Maes, 1994]). These systems take the approach of trying to statistically match users with similar interests, compare their evaluations, and propose items to a user which other users with similar interest considered “good.” Such statistical systems effectively try to “compute interest groups”. This computation is a black box for the users, and they can only be surprised about how it comes about that something was proposed to them (and it takes corresponding trust into the designers of the statistical method having built in the kinds of preferences a specific user also would like to have). Moreover, a general problem with this statistical approach is that it is notoriously hard to evaluate whether it works or how it can work better.

Grassroots clearly takes a quite different approach there. It makes group and information flow structures visible to cooperative people, who then can understand what kinds of information they are getting from where and what they are missing out on. They can explicitly choose why they might want to do so for specific sources, and they can play what-if games quite easily, that is, they can browse for which kind of information they would get, if they were to create a certain subscription link. They also can get angry about specific individuals who do not live up to certain expectations; they can consider them (socially or even legally) liable for the kinds of outflow they create or didn't create. We believe that putting people into the picture as first-class citizens is especially advantageous when people are seriously expected to be or interested in being cooperating partners (e.g. as part of organizations).

## 3.3 Organizing People

Many conventional systems provide separate interfaces for manipulating data and for manipulating user groups for broadcasting or access control. Sometimes, only system administrators are allowed to manipulate such groups. In Grassroots, any user can create and manipulate both folders and rosters with almost the same interface based on cut, copy and paste. Not only the owner of the roster but all users permitted to read the roster can use it for broadcasting messages and authorizing members to access some collections.

We can not ignore functionality-performance trade off. Some advanced mechanizm (e.g. caching) could be used to keep the performance of authorization at an acceptable level.

**Representing a Hierarchy of Groups Using Rosters**

A user can describe a hierarchy of human groups by creating sub-rosters under rosters. The operations involved are the same as for organizing hierarchical collections of data using sub-folders. If a link points from one roster to another roster, the latter roster effectively works as the former's sub-roster when used for access control or collective communication. This means that a roster can work as a part of more than one hierarchy not only statically but also functionally.

As a result, in order to represent an organization of people, different users can provide rosters structured in different ways. The most useful and best maintained one would be used by many people.

**System-Maintained Rosters**

Some rosters are automatically created and maintained by the Grassroots system. One is the roster of all users registered in one server. It includes links to the registered users' root folders. This roster belongs to a special user whose name is *grassroots*. Another type of system-maintained roster is *subscribers*. When a collector is subscribed to by some users, a roster named *subscribers* is maintained under the collector and links to the subscribing folders (folders in which subscription links are stored) are kept in the roster. *Subscribers* can be seen as an automatically maintained list of people who are interested in the collector's topic. Similarly, rosters named *forwarders* are maintained under a collection.

## 4. User Interface of the Grassroots Prototype

In this section, we describe the user interface of the current prototype of Grassroots. The current Grassroots prototype is an http-proxy based system which allows users to interact using any Web browser. There are no client-side augmentations. For each new user, a Grassroots account has to be set up in advance. This is because Grassroots deals with authentication (and users have to log in initially).

The current implementation is a set of perl scripts running on a Unix CERN http proxy server. This proxy code intercepts browser requests in the usual way. It then either generates a reply when it is a request to the proxy itself, or in the usual other case it will just add a menubar to the top of the page which the user is viewing.

### 4.1 The Grassroots Menubar

Figure 10 shows an example page with the Grassroots menubar added near the top of the page. The Menubar includes menus such as *news*, *copy this location*, etc.

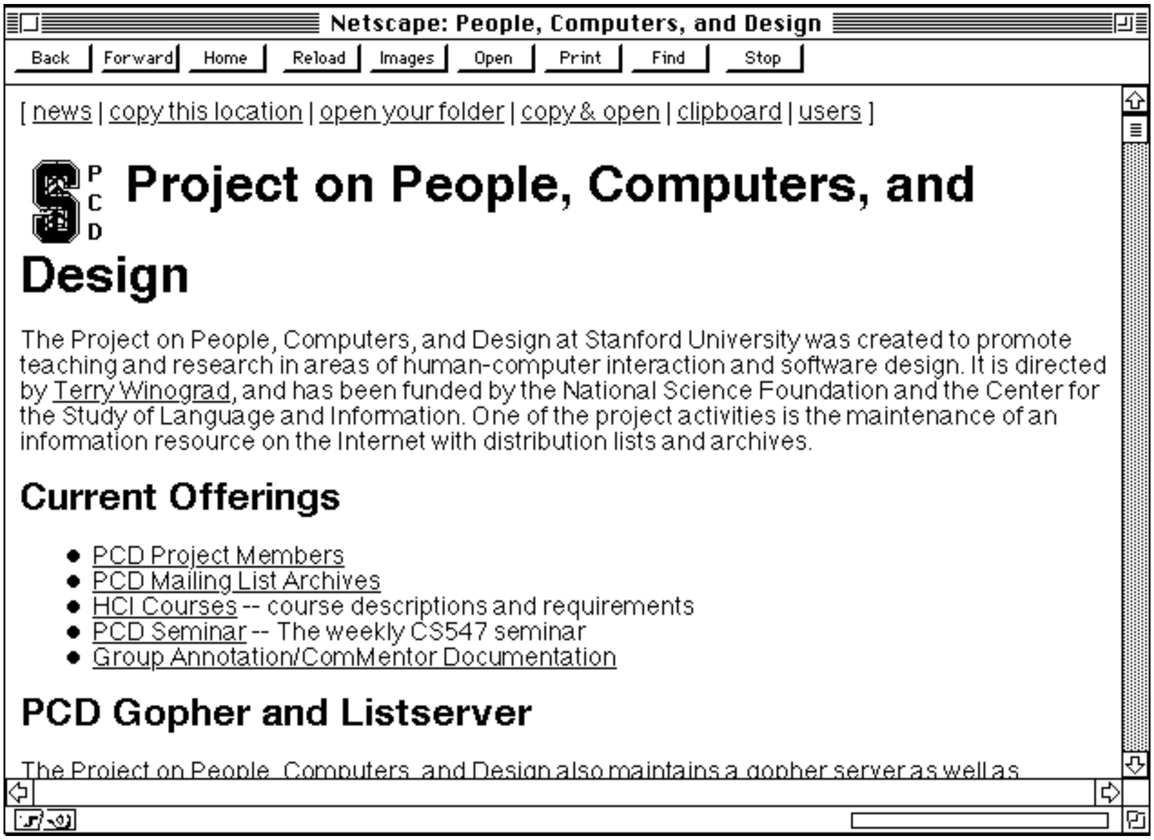



**Figure 10. Grassroots Menubar: Screenshots.**

The corresponding function of each button is described in Table 4.

| Button | Function Description |
|---|---|
| **news** | Open the notifier. |
| **copy this location** | Copy the URL of the browser's current page to the user's clipboard (of Grassroots). |
| **open your folder** | Open the root folder of the user. |
| **copy & open** | The shortcut for "copy this location"+"open your folder". |
| **clipboard** | Open the the user's own persistent clipboard which is stored in the proxy. The clipboard can contain multiple items. |
| **users** | Open the roster that lists all users on the server. |

**Table 4. Grassroots Menubar: Possible Actions.**

## 4.2 Collectors: Folders and Rosters

The user interface of a folder and a roster are shown in Figure 11 and Figure 12. The collector's path name is displayed in a large font at the top of the page. By clicking on any item in this path, the collectors at any level in the hierarchy can be opened. Below the path name, there is a row of buttons whose function is described in Table 5. The same row is also replicated for convenience at the bottom of the page. Though it is not desirable to have buttons in a document, we took this approach as a temporary solution to make the prototype accessible using currently wide-spread browsers.

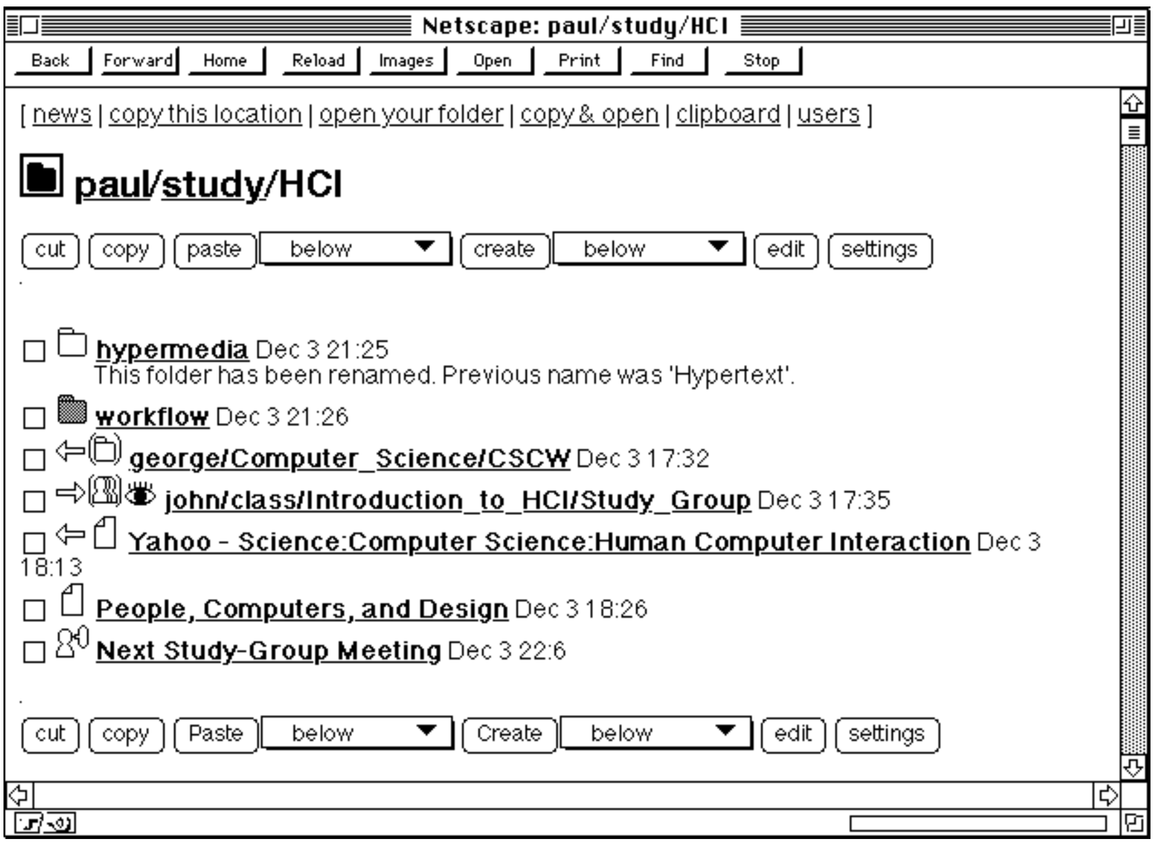



**Figure 11: A Folder: Screenshot**

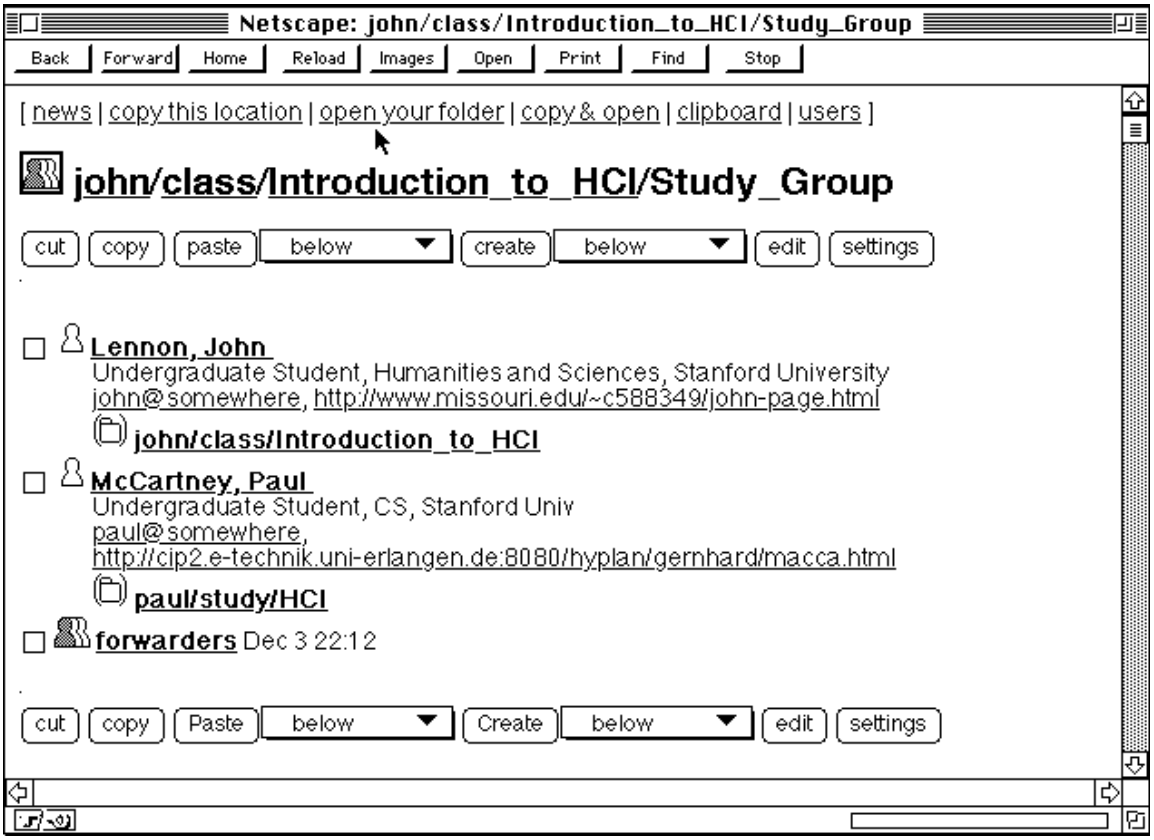



**Figure 12: A Roster: Screenshot**

The list of items in a collector are presented between the two rows of buttons. Folders present the list in the following way: At the left end of each line, there is a set of icons that describe the objects according to the scheme shown in Figure 13. Next to the icon is the name of the object (which is also an HTML anchor). If the object is a link, clicking on its name results in jumping to the location. If the object is a collector, it can be opened by clicking on the name. The date and time of the last update are also shown.

Rosters present the list in almost the same way except for links to folders. When a link to a folder is created or pasted into a roster, the user profile of the folder's owner is retrieved. With this user profile, a link to a folder is displayed in the scheme depicted in Figure 12.

| Button | Function Description |
|---|---|
| **cut** | Cut the selected objects into the user's clipboard.[3] |
| **copy** | Copy the selected objects into the user's clipboard. |
| **paste** | |
| below | Paste the objects on the clipboard below the selected object. [4] |
| into | Paste the objects on the clipboard into the selected folders.[5] |
| message | Post the objects on the clipboard to the selected objects.[6] |
| **create** | |
| below | Show the dialog for creating an object and paste the created object below the selected object.[7] |
| into | Show the dialog for creating an object and paste the created object into the selected folders .[8] |
| message | Show the dialog for creating an object and post the created object to the selected objects.[9] |
| **edit** | show the dialog to change the attributes of the selected object(*) |
| **setting** | show the dialog to set the special attributes of the selected links such as subscription, forward, read authorization or write authorization. |

**Table 5. Collector: Possible Actions.**

[3] Each Grassroots user has a persistent clipboard in the proxy server where the user account is registered. The clipboard can keep multiple items.

[4] If nothing is selected, the objects will be pasted at the top. If multiple objects are selected, the top one is considered "the selected object".

[5] The objects will appear as the top items of the selected collectors.

[6] If nothing is selected, the objects will be posted to the currently open collector. If links to collectors are selected, the objects will be posted to the collectors pointed by these links.

[7] Dito. Cf. the footnote for the corresponding paste attribute.

[8] Dito. Cf. the footnote for the corresponding paste attribute.

[9] Dito. Cf. the footnote for the corresponding paste attribute.

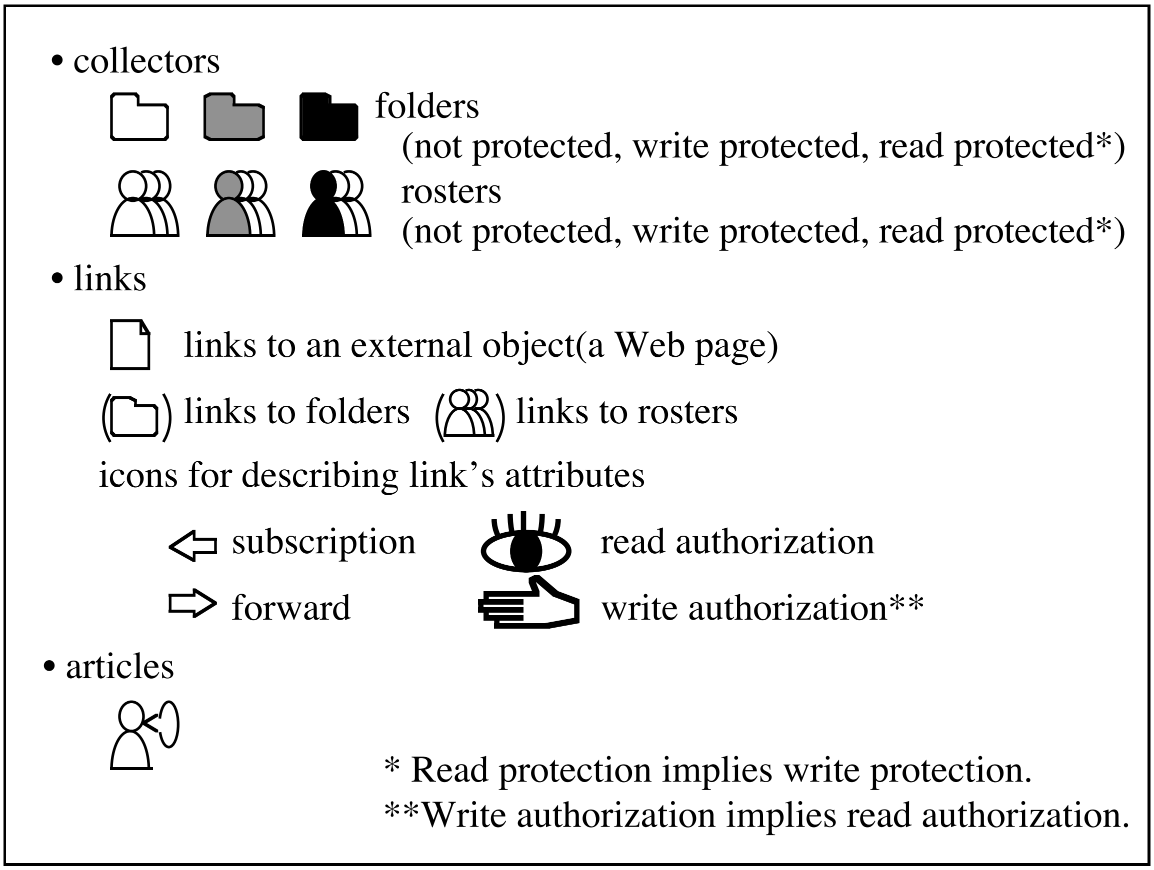



**Figure 13. User Interface Icons**

### 4.3 The Notifier

An image of a notifier is shown in Figure 14. In each section, the name of the corresponding folder appears on the top and serves as a heading.

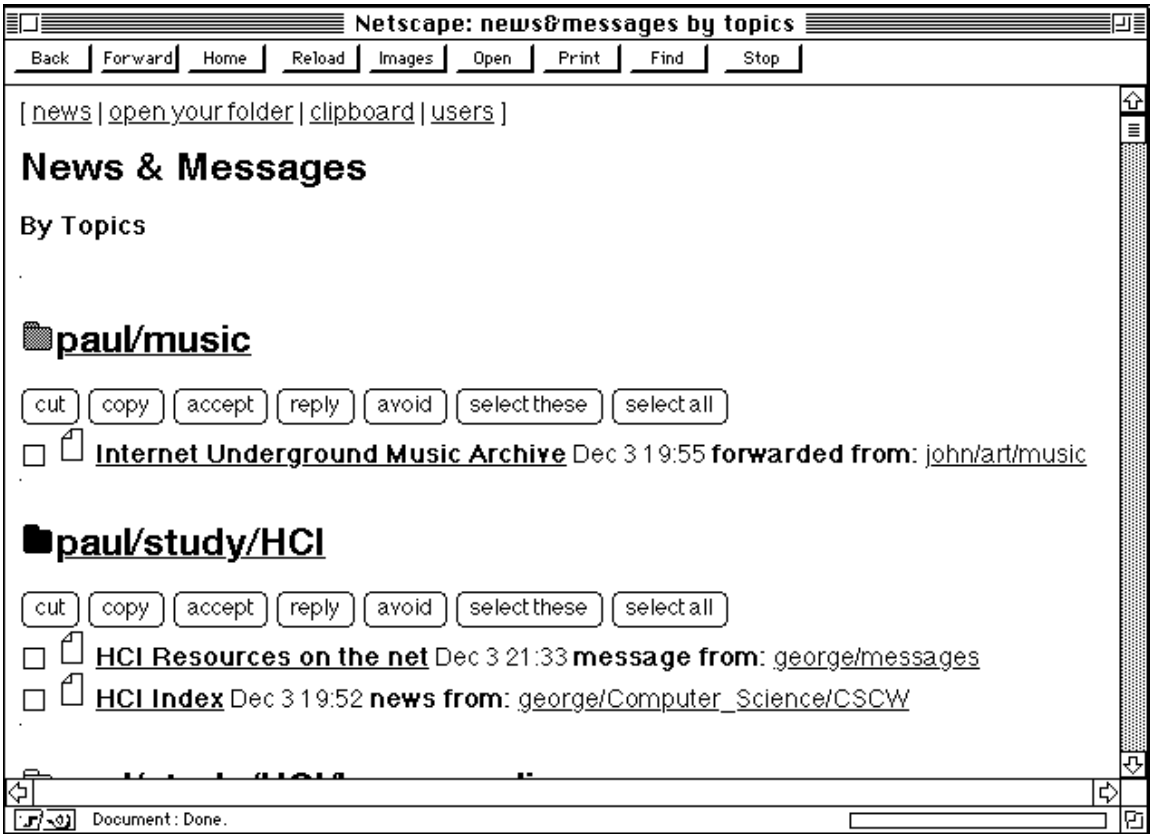



**Figure 14: Notifier: Screenshot.**

Below the folder name, there is a row of buttons whose function is described in Table 6. Below these buttons is the list of objects in this section. The format of the list is similar to the one in the folder. If an item has arrived because of subscription to folder F, the line indicates “news from F”. Similarly, if an item has been forwarded from folder F or posted from folder F, the line indicates “forwarded from F” or “message from F” with “F” being an anchor with the URL of folder F.

| Button | Function Description |
|---|---|
| **cut** | Cut the selected objects into the user's clipboard. |
| **copy** | Copy the selected objects into the user's clipboard. |
| **accept** | Move the selected objects into the collector that corresponds to the section of the notifier. |
| **reply** | Show a dialog for sending a reply. |
| **avoid** | Move the selected objects into a special folder so that those objects will not arrive in the notifier again. |
| **select these** | Select all objects in the section. |
| **select all** | Select all objects in the notifier. |

**Table 6. Notifier: Possible Actions.**

The summary of the notifier is also available. Its image is shown in Figure 15.

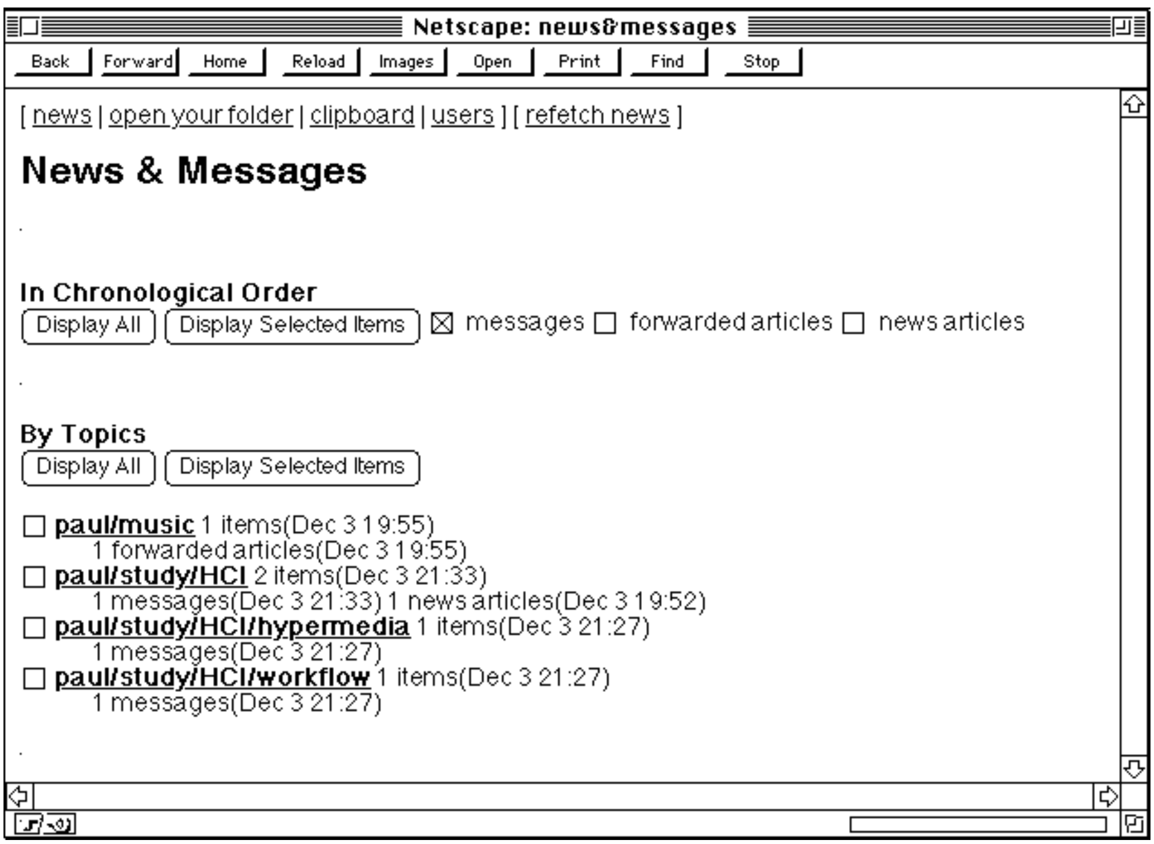



**Figure 15. Notifier Summary Screen: Screenshot**

## 5. Interaction with External Systems

Grassroots users can interact with other systems through Grassroots. Even if no-one else uses Grassroots, a user can benefit from using it. The interaction with other systems also allows users of other systems to get involved in the collaborations taking place in Grassroots. We describe here two examples.

**Subscription to Web pages**

Grassroots allows a user to have a subscription link to any Web page. All links (HTML anchors) newly added to the page will be brought to the subscriber's notifier. A page consisting of a list of links is a good target for subscription. Subscription to a hypermail archive brings the subscriber new messages that have appeared on the archive page. This effectively allows a user to get on or off the mailing list without bothering its administrator. A page of Yahoo is also a good target for subscription, because it consists of a list of links gathered under one topic. Subscription to UseNet News is also possible since newsgroups on the Web are lists of links to the articles. If a user sets subscription links from her folders to these pages, all the links added to these pages will appear in the appropriate section of the user's notifier.

**External Users on Rosters**

In a roster, a user can create an entry representing an external "virtual" user. An entry representing an external user consists of the user's e-mail address, home page URL, and some profile information. When a roster is subscribed to, the external user's home page is subscribed. When someone posts or forwards articles to the roster, it will be sent to the external user as well, in the form of an e-mail.

## 6. Experimental Feedback

Grassroots has been in experimental use by a small number of users for about three months. We do not yet have significant data from actual usage of the system, but we have a number of qualitative observations which we can already point out at this preliminary stage.

Overall, testers generally report a very favorable impression. In particular, the uniform way Grassroots deals with notification is highly favored presumably a need filled by a gap created by the absence of anything like this in the current Web. Users report having gained back a sense of control on information inflow; for instance, the ability of resting assured that Grassroots will check events and notify appropriately, for instance, when certain conference information is provided, etc. The uniform notification mechanism especially enables a number of usages which are possible but often not practiced in the current Web. This includes "subscriptions" to on-line newsletters available on the Web. Without Grassroots (or e-mail notification) few people are persistent enough to check a set of sites again and again for the possibility of there being a new issue of a newsletter or such. It can already be seen how Grassroots users keep easily track of a larger number of newsletters, and they find it edifying to see the new material right on the day of its appearance.

Common usages currently include pointing Grassroots to hypermail archives of mailing lists or to a selected set of Yahoo pages or certain other index pages for which a user would like to observe changes. (This current emphasis is of course because there are not yet many other users to whose folders one could point. But this point also shows that Grassroots can already be usefully used without having to attain a critical mass of users.)

Another effect seems to be that people more clearly think about which kinds of information to receive in pull mode, and which kinds in push mode. For example, with a hypermail archive declared as a Grassroots collector, there is virtually no difference any more between reading a mailing list via an e-mail subscription (possibly with messages automatically filtered into a special folder) or via the Grassroots notifier (with a link set onto a hypermail). This gives additional flexibility.

Although technically possible to read also one's personal e-mail in Grassroots, the current prototype implementation seems not to be sufficiently fast relative to a highly customized e-mail environment such that users generally still keep their e-mailboxes in the usual separate way. However, a difference which we already can see in the initial test use is that users will generally move much of their e-mail traffic (from mailing lists, etc.) to Grassroots, and gain back control over their e-mailbox.

Since January 1996, a CSCW course in the Computer Science Department of Stanford University, which consists of 30 students, has been using Grassroots as the main media of communication and the basis of a virtual community.

## 7. Related Work

Because Grassroots unifies e-mail, news, and shared hotlists, any previous work involving them are related with Grassroots. Here we choose some of such work and compare them with Grassroots.

To achieve the same purpose of categorizing incoming mail into folders, Grassroots and Information Lens [Malone, 1987] take different approaches. Information Lens uses rules to categorize incoming mail into folders. In Grassroots each folder has an address and messages can be categorized by the senders. Grassroots resembles gIBIS [Conklin, 1988] in that both let users create "issue(topic)-based" collections of information and link them together. While gIBIS is tuned to design discussions and requires a proprietary browser, Grassroots is a general purpose system and is accessible with any Web browser. Yahoo provides shared hotlists in a topic-based hierarchy. While the Yahoo system does not allow an individual user to change the hierarchical hotlist from the browser's screen, Grassroots lets users structure their own hierarchies of hotlists from any Web browser. In Grassroots, mailboxes, newsgroups, and hotlists are united into one type of object.

Grassroots can also be considered a tailorable tool for cooperative work. By combining components of Grassroots, end-users can form a large variety of functionalities for supporting cooperative work. However, its level of abstraction is different from that of Oval [Malone, 1992]. Grassroots' building blocks (collectors, notifiers, links and articles) are at a higher level than those of Oval. This means the end-users are required to do less work to achieve certain functionalities. It also means that generality of Grassroots is lower than that of Oval. Grassroots could fit somewhere between Oval, a "radically" tailorable tool, and single purpose conventional system such as e-mail, news or shared hotlists.

## 8. Conclusion

Grassroots uniformly integrates functionalities of many existing systems such as e-mail, news and shared hotlists in one simple user-conceptual model. Despite the prototype stage of the implementation, it already adds value for its test users in daily usage.

In particular, we see Grassroots demonstrating a clear need for a uniform mechanism of notification, that is, something which is not available in the current infrastructure including e-mail, Usenet News, and World-Wide Web. Grassroots identifies notification usages and provides a framework for how to understand them. It also shows how a uniform mechanism might be attained in a next-generation version of the World-Wide Web. For example, we can supplant current Web concepts such as its notion of a link with concepts from Grassroots such as its various attributes of links like subscription and forward.

Grassroots can thus be seen as the communication medium of the 21st century.

**Acknowledgments**

We would like to express our sincere thanks to all people who helped and gave advice to Grassroots project, used the system as experimental users, encouraged the system development, and inspired the original ideas.

**References**

Ackerman, M.S. (1994). Augmenting the Organizational Memory: A Field Study of Answer Garden. *Proceedings of Computer-Supported Cooperative Work (CSCW 94),* pp. 243-252.

Agre, P. (1995). Community and Democracy. *The Network Observer*, July. Availble at URL http://communication.ucsd.edu/pagre/tno/july-1995.html#community.

Conklin, J. and Begeman, M. L. (1988). gIBIS: A Hypertext Tool for Exploratory Policy Discussion. *Proceedings of Computer-Supported Cooperative Work (CSCW88)*, pp. 140-152.

Conklin, E. J. (1992). Capturing Organizational Memory. *Groupware '92*. Morgan Kaufmann Publishers.

Dourish, P., and V. Bellotti (1992). Awareness and Coordination in Shared Workspaces. *Proceedings of Computer-Supported Cooperative Work (CSCW 92),* pp. 107-114.

Goldberg, D., Nichols, D., Oki, B.M., and Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM* (Dec. 1992) vol.35, no.12, pp. 61-70.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J., (1994). GroupLens: An Open Architecture for Collaborative Filtering of Netnews. *Proceedings of Computer-Supported Cooperative Work (CSCW 94)*, pp. 175-186.

Maes, P. (1994). Agents that Reduce Work and Information Overload. *Communications of the ACM* (Jul. 1994) vol.37, no. 7, pp. 31-40.

Malone, T.W., Grant, K. R., Lai, K. Y., Rao, R. and Rosenblitt, D. (1987). Semistructured messages are surprisingly useful for computer-supported coordination. *ACM Transactions on Office Information Systems*. 5 1987, 115-131.

Malone, T. W., Lai, K. Y., and Fry, C. (1992). Experiments with Oval: A Radically Tailorable Tool for Cooperative Work. *Proceedings of Computer-Supported Cooperative Work(CSCW92)*, pp. 289-297.

URL-minder. Availble at URL http://www.netmind.com/URL-minder/URL-minder.html

Yahoo. Available at URL http://www.yahoo.com/