

# Social Task Networks: Personal and Collaborative Task Formulation and Management in Social Networking Sites

Yolanda Gil, Paul Groth, and Varun Ratnakar

Information Sciences Institute  
University of Southern California  
[gil@isi.edu](mailto:gil@isi.edu), [pgroth@few.vu.nl](mailto:pgroth@few.vu.nl), [varunr@isi.edu](mailto:varunr@isi.edu)

## Abstract

Personal tasks are managed with a variety of mechanisms from To-Do lists to calendars to emails. Task management remains challenging, since many tasks are interrelated, some may depend on other people's tasks to be accomplished, and their priority and status changes over time. While many tasks could be automated by services and agents available on the Web, To-Do lists often capture tasks in textual form that combine non-automatable aspects. We have designed a new approach that combines techniques for task representations in hierarchical planning and for sharing in social networking sites. We broaden the notion of assistance so it is not just provided by computers but also by other people.

## Introduction

Personal tasks are managed with a variety of mechanisms from To-Do lists to calendars to emails. Several systems have focused on providing task assistance to users based on the content of their email and calendar [Shen et al. 2006; Freed et al 2008]. To-Do lists have been found to be the most popular personal information management tools, used by more than 60% of people [Jones and Thomas 1997]. Yet there is no automated system to interpret and act upon them when appropriate on behalf of the user. While many personal tasks could be automated, user To-Dos are often captured in textual form that combine personal non-automatable aspects.

This paper describes our work in designing an assistant for To-Do lists. Our journey began by collecting To-Do data in a controlled setting. After understanding the particular characteristics of To-Do lists, we designed an approach to interpret them and map them to automated capabilities available in the user's environment. We then moved to a setting where users were entering To-Do lists in the wild through an application developed for a social networking site. This experience led us to a set of design desiderata that we realized as a To-Do management tool for collaborative task management. We arrived at the notion of *Social Task Networks*, which intertwine user tasks among themselves, with the user's social network,

and with web resources to create a rich tapestry of what personal tasks are all about.

The paper reports on data collected from users of progressively more sophisticated versions of our software. The data collected and the experiments performed were meant to inform our research, rather than prove conclusively the significance of user behaviors.

## Personal To-Do Lists

Our initial goal was to process To-Do list entries and map them to tasks that can be automated for the user by a set of agents. To understand what would be required to interpret To-Do lists, we collected and analyzed To-Do entries from users.

We collected a corpus of 2400 To-Do list entries from a dozen users. These were users of the CALO personal assistant [Myers et al. 2007] during two subsequent annual evaluation periods that lasted several months each. The system included a simple user interface to manage To-Do items. A subset of 300 entries were extracted as a reference corpus and used for development and analysis purposes. The remainder 2100 entries were set aside for evaluation.

Our analysis showed that to-do lists have idiosyncratic content that make their interpretation a challenge [Gil and Ratnakar 2008a]. Less than one in seven entries could be automated by some agent, since many entries contain tasks that only the user can do. Of those, most were incomplete and did not specify necessary arguments. Two thirds of the entries in that study did not begin with a verb. Many do not specify the action to be performed. More details of this analysis can be found in [Gil and Ratnakar 2008a].

In summary, to-do list entries are not well-formed or complete sentences. They lack the dialogue context present in conversational interfaces for agent tasking.

## Automating To-Do Lists in an Agent-Based System

Since our initial analysis revealed that the format of To-Do entries is not very amenable to natural language processing tools that can parse and create a structured interpretation, we developed an approach that exploit paraphrases of the target tasks that the agents can perform

and that specify how the free-text maps to the task arguments [Gil and Ratnakar 2008b]. As users manually assign To-Do to agents for automation, the system improves its performance by learning new paraphrases. We implemented this approach in a To-Do interpretation system called Beamer, and evaluated it with the corpus that we had collected as described in the previous section.

Beamer was found to have 86.7% accuracy in determining whether a to-do list entry is relevant to any of the agent capabilities, over 90% correctness in finding the agent capability that is appropriate for automating a given to-do entry, and very high correctness in determining which chunks of the to-do entry text correspond to which arguments in the target task so that only 0.2 to 0.4 edits on average are required from a user for a given to-do entry.

There was a limited amount of agents available to automate To-Do entries. Therefore, we moved on to explore possibilities for automation provided on the Web.

## Personal To-Do Lists on the Web

We wanted to target the Web as a substrate for automation. For example, we would link user To-Do entries to web resources that would be helpful with their tasks. We also wanted to analyze To-Dos in a real setting, since our corpus had been collected in an experimental setting. To this end, we developed a To-Do List application for a popular social networking site, since users keep a lot of personal information in it and visit it constantly so we thought it would be a convenient substrate to investigate personal task management tools. We also thought that eventually the applications for the site would themselves be able to provide automated capabilities.

The system we developed is freely available<sup>1</sup>. The application is instrumented to collect data from actual use and to facilitate integration with Beamer.

We did an analysis based on the initial user group for this application. The application is still in use today and collecting additional data. We analyzed 1500 To-Do items collected from 325 people. Of those people, 100 were stable users of the application.

This prototype revealed a great deal about the kinds of tasks that people jot in their To-Do lists, and about the potential for automation. Some observations are:

- *Many tasks had very coarse granularity.* Examples are “Get Christmas presents”, “Study nursing”, and “Get back in shape”. These are high-level tasks that involve many substeps and activities. There are no concrete first steps enumerated, which would be useful for a user to get started on the overall goal expressed in the To-Do.

- *Many tasks that were concrete had only some aspects that could be automated.* For example, “Write Christmas cards” or “Read Dune” could involve some on-line purchasing that could be automated but most of the activity was meant to be done by the person herself.
- *Some tasks could be fully or mostly automated.* Examples include “Renew my driver’s license”, “Buy iPhone”, “Rent Aliens movie”.
- *Many tasks were not amenable to automation.* Examples include “Be a true Christian” and “Go to the mall”.
- *Many entries were in other languages.* Even though the application had all its labels and titles in English, we found that many users adopted it to jot To-Dos in their own vernacular.

We noted that many To-Do entries referred to web resources. For example, an entry for “Buy Borges book” would include a URI for a bookseller’s web site.

## Automating To-Do Tasks on the Web

Based on the observations above, we revisited the design of our To-Do application. The following were design desiderata for our new system:

- To-Do lists should be hierarchical and describe tasks in terms of their constituent subtasks. The set of To-Dos that are active at any given time are often part of enveloping tasks. This will allow users to express tasks at coarser and finer levels of granularity, giving the task a high level coarser description when it is first jotted down and drilling down into details when the task is being pursued.
- To-Do lists should include both automatable steps and non-automatable steps. The latter kind is important to track because it provides context for the user and because it serves as a reminder that the task is pending their attention and up to them and not the system.
- Automation could be done by different systems, but the interface to the To-Do list should be the same. Some tasks could be automated by on-line scripting applications, such as IBM’s Co-Scripter, Mozilla’s iMacros, or Yahoo Pipes. The user interface for To-Dos should be unchanged, even though the underlying execution engine is widely different.

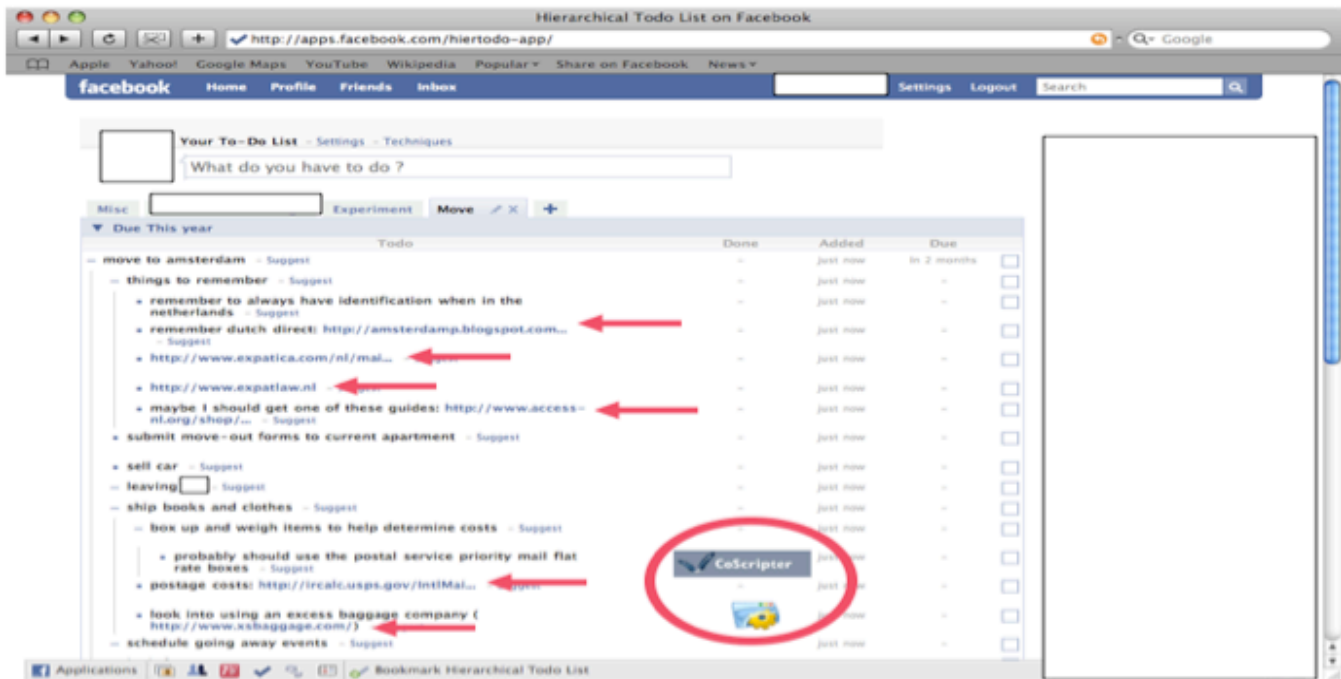
We developed a hierarchical To-Do list application for Facebook, the popular social networking site. Figure 1 shows a snapshot of the user interface, illustrating the use of pointers to web resources as URIs as well as web scripts. The system is freely available<sup>2</sup>.

---

<sup>1</sup> <http://apps.facebook.com/todo-lists/>

---

<sup>2</sup> <http://apps.facebook.com/hiertodo-app/>



**Figure 1. Hierarchical To-Do List Manager, where some tasks must be done by the user while others are to be automated by web scripts or agents. Notice that many tasks include pointers to web resources as URIs as well as web scripts.**

An important thing that we learned with this tool was regarding mechanisms to automate user To-Dos. Initially we had planned to use Facebook “apps”, modular applications built for Facebook users. For example, there are apps for calendar management and for sending emails and messages. Although these apps could in principle be used to automate tasks for users, we found that the apps that we thought would be relevant for automation did not have the appropriate hooks for us to integrate with.

In seeing tasks stated at a finer granularity, it became clear that at some point in the decomposition tasks could be done in collaboration with others. We investigated this in a subsequent version of the system.

## Collaborative To-Do Management in Social Networking Sites

Finding web scripts and agents that could assist the user in accomplishing their To-Dos is useful but only a limited subset of To-Dos could be addressed in this manner. For many To-Do entries, finding other users that can assist in some way reflects the way many To-Dos are accomplished in practice. There are many opportunities for individuals within a social network to provide help with tasks, such as delegating, collaborating, and providing know-how about accomplishing tasks [Gil et al 2009]. Therefore, we added the following desiderata:

- To-Dos should be exportable so that assistance in terms of automation can be provided by other

individuals in the social network. For example, a project assistant may provide the maximum allowable amount to spend in a new laptop purchase, which may be just one step in the overall task of getting a new computer. Other users can decline the task, but if they accept the user can have visibility on its progress and status.

- To-Dos should be shareable so that assistance in terms of know-how may be provided by other individuals in the social network. For example, if a To-Do entry is to find a hotel in Paris, someone else may have a list of favorites that they are willing to share.
- To-Do decompositions should be shareable, so that know-how can be shared. For example, if someone is looking for job announcements someone else may have just looked and have a task description to share: a set of steps that they followed to find job announcements in diverse web sites and mailing different individuals.
- Assistance could be provided by matching To-Do entries with automated applications and with other user’s To-Do decompositions through paraphrasing through Beamer.

We developed a new application based on our previous versions that incorporates these new desiderata for sharing. The application uses a social networking site as a substrate for communication and collaboration. We describe its features in the rest of this section, and then discuss some initial feedback that we received from users in a small controlled experiment.



Figure 2. Users can post their To-Dos to their social network and get help from other users.

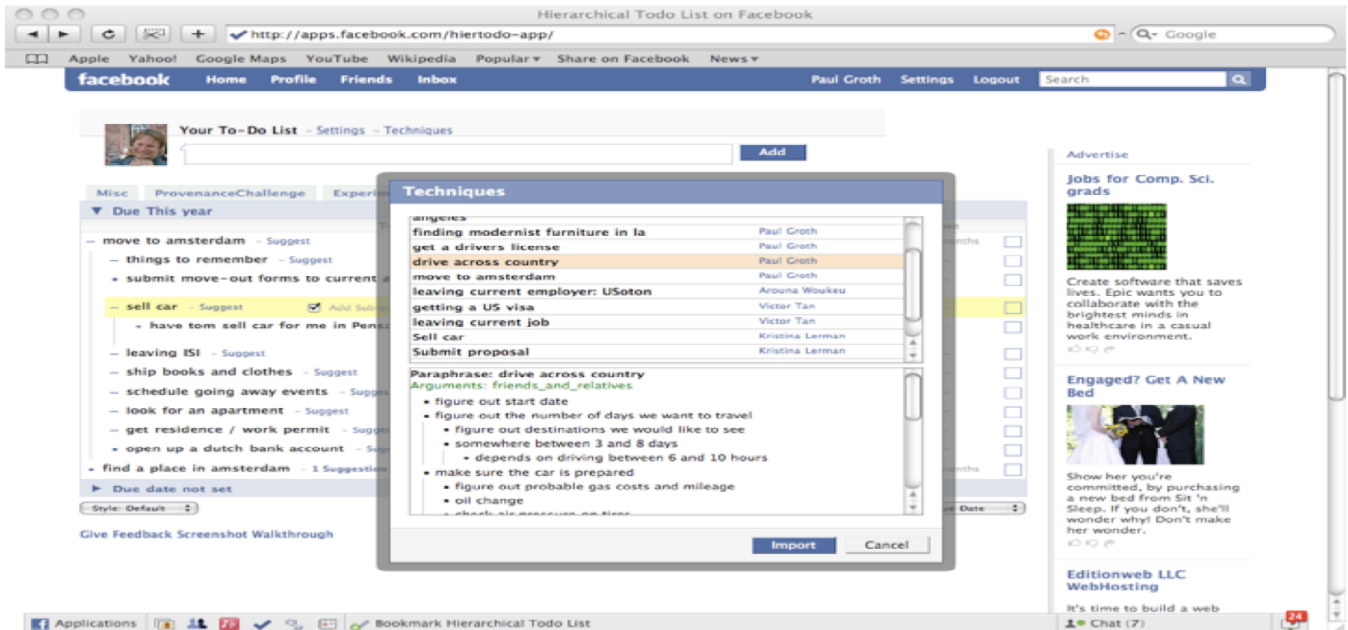


Figure 3. Users can share their “know-how” as techniques that are reusable by other users.

The latest version of the application provides standard To-Do features that our baseline application had. At the top of the application, there is an entry box for entering To-Do items, modeled after the status update box in Facebook. Each To-Do item can be assigned a priority, category, due date and be marked as completed. Users can organize their To-Do items into personal categories using tabs. The application also includes features from the hierarchical version of the software to allow users to organize To-Do items in a hierarchical fashion to arbitrary depth.

The Hierarchical To-Do List provides an interface for managing tasks in the context of a user’s social network. By embedding it within a social networking site, the application has complete access to an explicit social graph. Furthermore, it is in an environment that users are comfortable with and use often. Therefore, users can be assisted with their To-Dos either by other users, by textual instructions on how to accomplish a task, or by automated procedures (most likely shared web scripts). Figure 2 shows this kind of assistance, where the user shared a To-

Do item “Move to Amsterdam” and another user in their social network responded with suggestions.

When a To-Do is accomplished, users can share the process they followed with others (think of this as sharing procedures just like you would share pictures). Inspired by sites such as eHow.com, we conjecture that users are willing to share their know-how. To enable users to share their To-Do lists with other users in a reusable fashion, we introduced the notion of techniques. A technique is a parameterized hierarchical To-Do list that is publicly available. A user can export from the Hierarchical To-Do Application a portion of their To-Do list as a technique. Likewise, users can import a technique into their own To-Do list. Figure 3 shows a list of techniques that are available in the system. In the figure, the technique “drive across country” is selected. Note, at the top of the technique an argument (friends and relatives) is listed. When exporting the technique, the user selected a set of words that were anonymized as parameters/arguments. When another user imports the technique, they will be asked to fill in the argument. Furthermore, the system keeps track of the provenance of To-Dos created from importing techniques. Users are notified when creating a technique that it will be completely public and available generally on the Web.

These investigations lead us to propose a new paradigm for To-Do management, which we present next.

## Social Task Networks

We define *Social Task Networks* (STNs) as tasks that are created and shared across users in a social network. The name also reflects that the structure we have given the To-Do list is as a decomposition much in the *hierarchical task networks* (HTN) style of AI planning.

An important feature of STNs is that they capture *task networks* through the following mechanisms:

- **networked subtasks** through links from tasks to subtasks in a hierarchy
- **networked web tasks** through links from tasks to web resources expressed as URIs
- **networks of user tasks** through links from a user to their personal tasks

Another important feature of STNs is that they capture *social tasks* through the following mechanisms:

- **networks of user tasks to other users** through links from a user’s shared task to other users that volunteer to provide assistance
- **networks of user tasks to other user tasks** through links between shared and reused techniques

STNs reflect the role of users personal tasks in their social fabric, and capture the role of web resources in the accomplishment of tasks.

## User Experiences with STNs

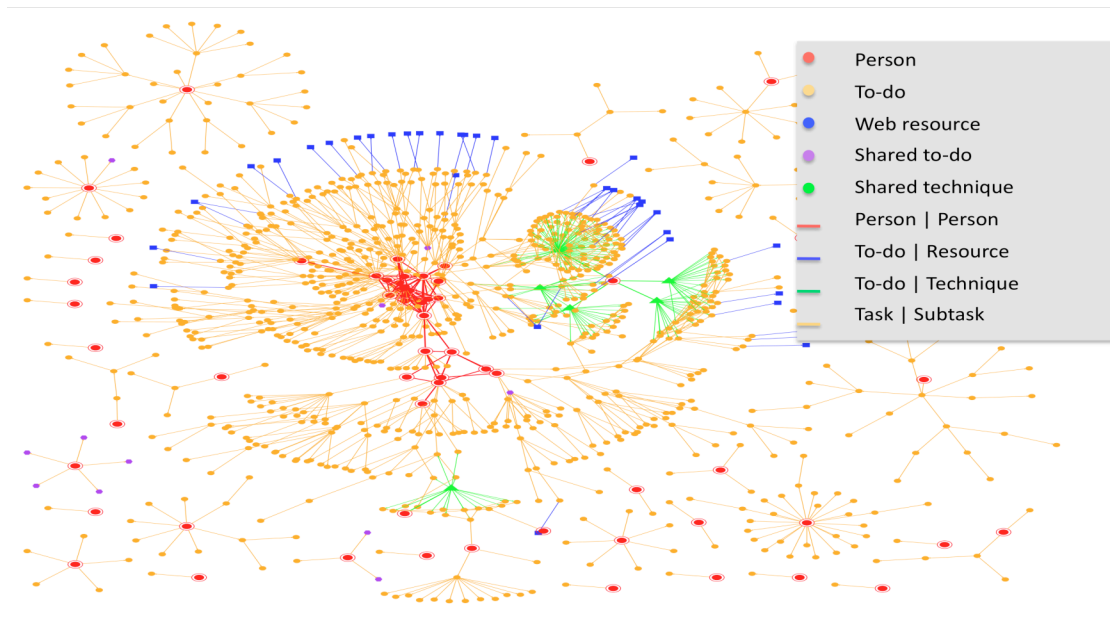
To gain an initial insight into how a social network of users might interact with the STN paradigm, we asked two groups of users to make use of this version of the application. The first group of users (Group A) consisted of 28 users and they were asked to perform the same task while using the application. All users had prior experience with the social networking site. We asked the user to perform the following activity for three days to start and then use the application further if it was useful to them:

*Imagine that you are planning a move to Los Angeles. Use the application, to write a detailed To-Do list of the things you would need to get done to achieve this goal.*

We gave a brief guide to these users highlighting the various features of the application. Since this was an artificial task, we followed up with reminder emails to all the users encouraging participation over the course of the initial three-day period. The second group of users (Group B) consisted of 5 new students and the project assistant involved introducing them to our institute. We believed that because of their many shared and similar tasks, they would be good candidates for using the application. In particular, we thought this might allow the project assistant to expend less resources in answering the same questions repeatedly.

These groups of users provided useful insights into how people might prefer to manage their tasks. First, we learned that people are very concerned about the privacy of their To-Dos even in their social network. They prefer to be very selective regarding who sees their To-Dos. Perhaps this kind of information is more personal than other things that are shared in social networking sites. Second, we found that users were more inclined to resort to face-to-face interactions rather than using the system. This might be due to the proximity of the users in physical space, being in the same office building. Finally, it is clear that without having significant help from the system the users will not be inclined to jot a lot of To-Dos or to be very specific about them. As we mentioned above, the more the system will help them the more we expect users will be enticed to jot To-Dos in a more specific and shareable manner.

Figure 4 depicts the Social Task Network of the user experiment just described. It shows users, their to-dos, and web resources, and the links among them. Each top-level To-Do item (i.e. one without a parent) is connected to the user that created it. Users are connected to their friends by red lines representing the social network. Web resources (URIs) are connected to each other through the social and task networks.



**Figure 4.** A visualization of a Social Task Network, showing links from people to tasks of others in their social network.

## Conclusions

In this paper, we introduced Social Task Networks, a novel approach to task management that makes transparent the relationships among users' tasks, their social network, and relevant web resources. An implementation of this approach integrated with a social networking site was described. Through observing the usage of the application, we gathered evidence that real world To-Do items are amenable to assistance from intelligent agents when tasks are properly decomposed and advertised. Because users are not accustomed to assistance and sharing for To-Dos, we plan to make the system more proactive when it recognizes that similar tasks were decomposed, automated, or collaboratively solved previously by others.

**Acknowledgements.** We would like to thank Karen Myers, Michael Freed, and Dustin Moskovitz for their useful feedback on this work. This research was supported in part by the Defense Advanced Research Projects Agency with grant HR0011-07-C-0060 and by the National Science Foundation with grant IIS-0948429.

## References

- Bellotti, V., B. Dalal, N. Good, P. Flynn, D. Bobrow, N. Ducheneaut. "What a To-do: Studies of Task Management Towards the Design of a Personal Task List Manager." ACM Conference on Human Factors in Computing Systems (CHI), 2004.
- Freed, M., Carbonell, J.G., Gordon, G.J., Hayes, J., Myers, B.A., Siewiorek, D.P., Smith, S., Steinfeld, A. and A.

Tomasic. "RADAR: A Personal Assistant that Learns to Reduce Email Overload." Proceedings of the Twenty-Third Conference of the Association for the Advancement of Artificial Intelligence (AAAI), 2008.

Gil, Y., and Ratnakar, V. "Towards Intelligent Assistance for To-Do Lists." ACM International Conference on Intelligent User Interfaces (IUI), 2008.

Gil, Y., and Ratnakar, V. "Automating To-Do Lists for Users: Interpretation of To-Dos for Selecting and Tasking Agents." Proceedings of the Twenty-Third Conference of the Association for the Advancement of Artificial Intelligence (AAAI), 2008.

Gil, Y., Groth, P. and V. Ratnakar. "Leveraging Social Networking Sites to Acquire Rich Task Structure." Proceedings of the IJCAI 2009 Workshop on User-Contributed Knowledge and Artificial Intelligence (WikiAI09), 2009.

Jones, S. R. and Thomas, P. J. 1997. "Empirical assessment of individuals' 'personal information management systems'." Behaviour and Information Technology 16(3): 158-160.

Myers, K., P. Berry, J. Blythe, K. Conley, M. Gervasio, D. McGuinness, D. Morley, A. Pfeffer, M. Pollack, and M. Tambe. "An Intelligent Personal Assistant for Task and Time Management." AI Magazine, 2007.

Shen, J., Li, L., Dietterich, T. G., and Herlocker, J. L. "A hybrid learning system for recognizing user tasks from desktop activities and email messages." Proceedings of the 11th International Conference on Intelligent User Interfaces, 2006.