Reminiscing a person's life from his lifelong todo list

Nicolas Kokkalis
Stanford University
353 Serra Mall
Stanford, CA 94305 USA
kokkalis@stanford.edu

Scott Klemmer
Stanford University
353 Serra Mall
Stanford, CA 94305 USA
srk@cs.stanford.edu

Mendel Rosenblum
Stanford University
353 Serra Mall
Stanford, CA 94305 USA
mendel@cs.stanford.edu

CHI 2011, May 7–12, 2011, Vancouver, BC, Canada.
ACM 978-1-4503-0268-5/11/05.

Abstract

To-do lists and project management software require their users to manually keep the lists current and relevant. Because users loath to do information gardening, these to-do lists can grow unwieldy and get cluttered with tasks that have lost their importance. Consequently, a common strategy is to periodically "declare bankruptcy" and start anew. This work seeks to create an intelligent task list that can be used for the entire lifetime of a person. Having a lifelong todo list is not only a great way to reminisce one's life but also it gives the structure to zoom in and out to important moments of it. A privileged few have Executive Assistants to organize their tasks but not everyone can afford this premium solution. Our methodology is to first understand where human assistants succeed yet software fails, and then codify these successful practices in an intelligent task list that can be used for life.

Keywords

Todo Lists, Task Management, Reminiscence

Introduction

Having a lifelong todo list is a great way for reminiscing a person's life experiences. Unfortunately, task lists suffer from clutter. This clutter often prevents users from using the same todo list for their entire life. One of the many reasons for the clutter is simply the fact that people typically wish to do way more tasks than what they can actually accomplish. Consequently, todo lists get filled up with too many tasks very quickly and become unmanageable. Users often react by starting brand new lists, sometimes, on different media or software. This behavior makes it very difficult to reconstruct a person's life from the tasks he accomplished through the years.

Some privileged people have executive or personal assistants to facilitate organization and maintenance of their personal or professional tasks. In contrast to how people treat software todo lists, in many cases people choose stick to the same human assistant for many years. We want to identify what are the abilities of great human assistants that make them perform so much better than any existing piece of software. The ultimate goal is to codify these successful practices in an intelligent task list – using machine learning and online workers to facilitate organization and maintenance of comparable if not better performance.

Human assistance generally takes two forms. First, an assistant can help organize task information and remind people when appropriate. (For example, remind a person to get milk when they are close to a grocery store.) Second, an assistant can help accomplish the task. (For example, buy the milk when needed.)

Let's consider another example: A person wants to register for CHI 2012. In this case, the assistant may not know what CHI 2012 is and their work may look as follows. First they perform an Internet search on the terms and discover that CHI is a conference, taking place in Austin, on May 5-10 and that the registration is not open yet. The assistant needs to perform several actions in response to this information. They mark "CHI 2012" as one of the contexts of the user and "Austin" as the user's predicted location for the period of May 5-11, 2012. Also, since the registration is not open yet, the assistant needs to schedule a reminder for themselves to check back the website every month until the registration opens. Additionally, the assistant must identify that this task falls under a template of tasks, say "Conference Participations". The template dictates more subtasks associated with this type of action. So the assistant, at a convenient time for the user, reconfirms that it is also needed to "book flights" and "book a hotel". At that time the assistant may be able to elicit more relevant tasks such as "Rent a car", and decide whether to update the "Conference Participations" template with these additional tasks. Eventually, when the time is right the assistant can execute as many of these tasks as possible and remind the user to accomplish the rest.

A fully automated computer system can be built to organize or execute tasks as simple as the milk example, but the is no system today that can fully automate more complex examples, such as organizing the CHI registration and trip. We believe that a combination of machine learning algorithms along with the use of online human workers has the potential to offer revolutionary solutions to this problem.

Other examples include breaking down tasks to smaller pieces, ordering tasks by importance and relevance, scheduling meetings, task delegation to user's peers, and administration of tasks shared between work groups, all of which are tasks practically handled by human assistants today.

Task Management accomplished by humans

So, what do successful human assistants do better than software, in managing someone's tasks? Our hypothesis is that human assistants success depends on the following 5 points.

1. Ability to *elicit more of the user's tasks* and make them feel confident that nothing will get lost or forgotten. This way the user has to communicate each desired task only once and ultimately never have to worry that it may go unattended.

2. Ability to *present the user only with the most urgent and relevant tasks* at each time. Generally the answer to a question "What do I have to do today" is a small list of things rather than a large corpus of all accumulated tasks. This question is currently a judgment call for the assistant to come up with a high quality selection of urgent and relevant tasks. The assistant needs to prioritize the user's tasks taking into consideration the user's current location and context, along with the personality and preferences of the user observed in previous occasions

3. Ability to assist the user in *starting off lingering tasks*. We call lingering a task that has been in a user's list for a long time. There are two types of such tasks. The ones that are important and need to get done and the ones that are lingering because they are simply unimportant or they lost their importance over time. A good assistant needs to distinguish between the two so that they can increase the urgency of the former and archive the latter.

   Once a lingering task is identified as important, the assistant needs to discover the reason it has not been accomplished. According to David Allen [2] the most important cause for a task to remain undone for a long time is because it seems too vague to the user. In this case, the assistant needs to help the user break down the task into smaller subtasks that require less activation energy to get started. For this break down, the assistant must use their judgment along with information that they specifically elicit from the user for this reason.

4. Ability to *execute some of the user's tasks and delegate others* to the user's peers or external professional services.

5. Ability to make the user feel that they *are still the one in control*, even though a big part of their life is now managed by others. All assistant decisions should be taken in complete transparency to the user and whenever thu user needs they should be able to quickly discover everything.

Task Management accomplished by a system

In the following paragraphs we describe in more detail what a computerized system can do to achieve a high quality outcome on each of these areas.

Task Elicitation

In order for a computer system to be able to elicit as many tasks as possible from the users it needs to try to lower the activation energy needed on the behalf of the users to add new tasks. For that reason, the system needs to be accessible everywhere, online and offline. It needs to be present on every device of the user, such as his computer, his smart phone, or his old landline phone. A natural language input interface is also helpful. This can be in the form of an online typed chat or a voice-recording button in the user's smartphone. As far as voice transcription is concerned, automatic voice transcription nowadays has achieved high levels of accuracy and for the fragments of speech that tools do not recognize with high confidence a crowdsourced workforce can be utilized. Also, the system can identify potential tasks sent to the user by her peers through email or through the system itself.

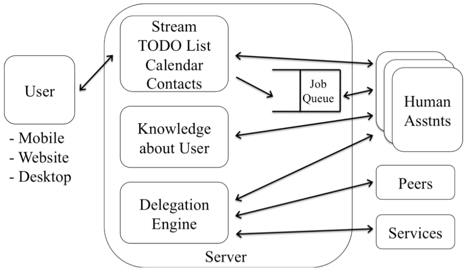

figure 1. The high level diagram of a computer system that uses the wizard of Oz approach to emulate an intelligent todo list and use it to discover what needs to be done to create it.

Urgent and relevant task selection

In order for the system to minimize the information overload that would be caused by an overwhelming number of tasks accumulated over time, the system needs to organize tasks by urgency and relevance to the user's current context. For this reason, the system needs to non-intrusively, over a long period of time, catalogue the user's frequent projects, locations, contexts, goals, and peers. At the same time, the system needs to maintain a variety of properties for each task. It needs to associate the task with projects, locations, contexts and goals; It needs to discover peer dependencies ("waiting-for") from both directions; It needs to assess and actively reevaluate the importance of the task; finally it needs to estimate the time needed to complete this task. Given all task properties and current context and location of the user the system should be able to evaluate only the relevant and important tasks.

Lingering task assistance

Computer systems are generally much better than humans in keeping track of the time a task has been sitting untouched in somebody's task list. Lingering tasks need to be re-evaluated over time. If a task initially had been assigned a low priority, the system needs to decide if the priority should be increased because the task has now a closer deadline. Also, if the task has high priority, the system needs to evaluate whether the task is not important anymore and consequently reduce it's priority or archive it. If a task has high priority and is still important the system needs to break it down to smaller subtasks. To achieve this the system can use information it elicits from a corpus of known cases automatically, or by prompting for the input of a human assistant, or by prompting the user him self at appropriate time. The system must also discover any psychological preconditions, context, or location needed for this task so that it can suggest the user to start working on it when these conditions apply in the future.

Execution and Delegation

There are multiple occasions where a user may assign a task to another person, who may forget to follow up on it. Obviously, a computerized system can keep track of all these task assignments and make sure they are followed up and reported as done when they are completed. Another situation where a system can be helpful is for tasks that can be executed automatically by the system or outsourced to human assistants or to a crowd of workers.

User's feeling of control

From the perspective of the user the system assists, manages, and organizes her tasks automatically. Regardless of whether these actions take place by software, human assistants, or a crowd, they need to be fully transparent to the user. The transparency needs to be maintained in a non-intrusive way that minimizes information overload to the user. The user should be able to easily identify what is going on for any of his tasks when he is searching for it. The user should be able to override any decision made by the system at any time. When such a user intervention happens the system must internally learn by its mistake. This learning can happen by recording the event, broadcasting it to all relevant assistants or penalizing the machine learning algorithms used to generate the mistake.

About the author attending the workshop

Nicolas Kokkalis is a Ph.D. student of HCI at Stanford University. From 2007 to 2010 he created web applications that attracted an aggregate of more than 20 million users. Nicolas has been especially interested in understanding what creates infectious user actions that lead to the explosive spread of certain websites on the Internet. His master's thesis was a fault tolerant distributed platform that solves and abstracts the problems of fault tolerance in mission critical servers. Nicolas designed and physically manufactured a motherboard with 8 FPGA PCI 64 cards and 2 Ethernet interfaces and wrote a linux device driver, as his undergraduate thesis. Nicolas's current research interests lie in solutions that proactively manage people's resources and amplify their capacities.

Citations

[1] Mark, G. The Supporting Role of Paper in Multi-tasking. Proc. CoPADD (2007)

[2] Allen, D. Getting Things Done. The Penguin Group (2001)