Sharedo: To-Do List Interface for Human-Agent Task Sharing

Jun Kato[†] **Daisuke Sakamoto**[‡] **Takeo Igarashi**[‡] **Masataka Goto**[†]
National Institute of Advanced Industrial Science and Technology[†], The University of Tokyo[‡]
{jun.kato, m.goto}@aist.go.jp[†], {d.sakamoto, takeo}@acm.org[‡]

ABSTRACT
In this paper, we propose a to-do list interface for sharing tasks between human and multiple agents including robots and software personal assistants. While much work on software architectures aims to achieve efficient (semi-)autonomous task coordination among human and agents, little work on user interfaces can be found for user-oriented flexible task coordination. Instead, most of the existing human-agent interfaces are designed to command a single agent to handle specific kinds of tasks. Meanwhile, our interface is designed to be a platform to share any kinds of tasks between users and multiple agents. When agents can handle the task, they ask for details and permission to execute it. Otherwise, they try supporting users or just keep silent. New tasks can be registered not only by humans but also by agents when errors occur that can only be fixed by human users. We present the interaction design and implementation of the interface, Sharedo, with three example agents, followed by brief user feedback collected from a preliminary user study.

Author Keywords
Human-robot interaction; social media platforms; to-do list.

ACM Classification Keywords
H.5.2. User Interfaces [Interaction styles]; I.2.9. Robotics [Commercial robots and applications].

INTRODUCTION
There is an increasing number of agents available in our daily lives, including software personal assistants on smartphones, e.g., Siri for iPhone and Cortana for Windows Phone, and robots at home, e.g. iRobot Roomba. While existing agents can handle various tasks through usable interfaces, e.g. speech and button-based interfaces, they are usually designed to instantly respond to the user. Therefore, to interact with such agents, the user needs to be prepared to clarify details. However, our daily lives are filled with ambiguity. We do not know what we want to eat tonight, who will cook it, whether to order books online or to buy them at a nearby bookstore, when we have time for watching videos recommended by friends, etc. Such ambiguity in tasks does not need to be resolved instantly, and we sometimes want to keep them ambiguous till the time comes.

In addition, which agent should handle a task is sometimes ambiguous since multiple agents can handle it in different ways. For instance, to watch a movie, a shopping robot can buy a Blu-ray disc at a movie store and a web-based agent can show it online. Therefore, it is important to think of interaction techniques for task coordination as well as task specification.

For organizing ambiguous tasks whose details (who, when, how to handle them) are yet to be decided, a to-do list has been a good tool for human users [1]. Kreifelts investigated its use for human-human task sharing [8]. There are also many web services that allow sharing to-do lists with others such as Remember The Milk [10]. With these existing uses of to-do list interfaces for task sharing in mind, we propose Sharedo (Figure 1), a web-based to-do list interface that supports the ambiguous state in human-agent task sharing. It extends the use of the to-do list interface not only for humans but also for agents and allows them to share, discuss, and complete tasks together.

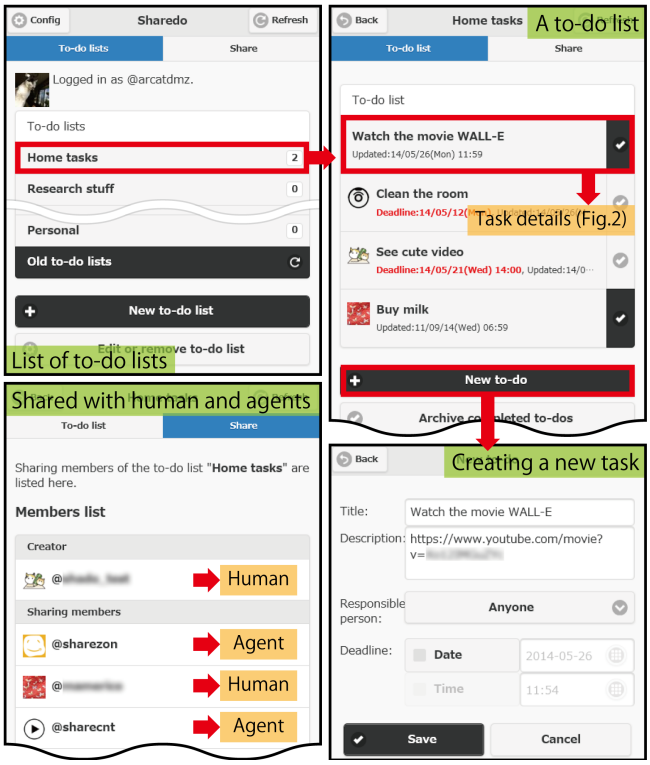

Figure 1. To-do list interface for human-agent task sharing

HAI 2014, October 29–31, 2014, Tsukuba, Japan.

ACM 978-1-4503-3035-0/14/10…$15.00.
http://dx.doi.org/10.1145/2658861.2658894

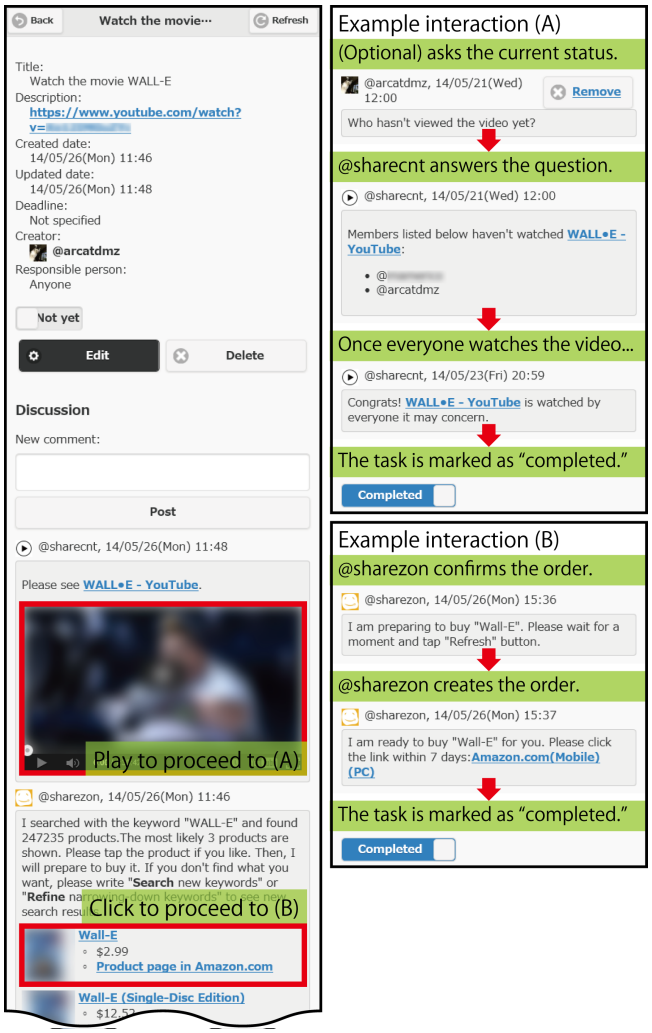

Figure 2. Discussion board for removing task ambiguity

All users and agents of the Sharedo interface have virtual embodiments associated with their social media accounts. Once they have logged-in to the platform by using the accounts, a to-do list can be created to share tasks with other users and agents. Both a user and agent can register new tasks to the list (the user through the web page and the agent through a web API). A task has its own page that shows its detailed information and a discussion board for discussing the details of a task, as shown in Figure 2. All users and agents are notified, whenever they edit the task information and create posts on the discussion board (the users through a message on the social media account and the agents through a web API hook). Discussion on the task gradually removes its ambiguity and leads to its completion. Finally, the task can be marked as completed either by a user or agent.

To provide concrete examples of human-agent interaction on the platform, three agents with different characteristics are implemented: a house cleaning agent that schedules the appropriate time for cleaning a room and executes it by controlling an iRobot Roomba robot, a shopping agent that suggests, purchases, and delivers products to the home with the help of the Amazon API, and a media content agent that helps in consuming media content such as reading online articles and watching online videos. In the following sections, related work, interaction design, implementation details, and user experience collected from a preliminary study are reported.

RELATED WORK

Software Architectures for Task Coordination

Much work has been done proposing software architectures and algorithms for autonomous task coordination of multi-agent systems [2], aiming at efficient task completion. A notable example is Electric Elves [3], in which human users and agents are equally represented as proxies to achieve human-agent collaboration. The agents in the system support human organization by tracking schedules and the locations of human users with palmtop computers, deciding presenters for meetings, and arranging food for the meetings by faxing orders to restaurants. While these agents consider user input, e.g., accepting/declining meeting schedules, and achieve adjustable autonomy, the task coordination is an autonomous process without user input. In more recent work [11], a prototype infrastructure was proposed that allows the user to help in task coordination, which resulted in more efficient robot-agent-person collaboration for disaster rescue.

These papers do provide some information on their graphical user interfaces, but their focus is consistently on software architectures and algorithms for efficient (semi-)autonomous task coordination. In contrast, our work focuses on a user interface design that creates an intuitive mapping between actions on shared to-do lists and events of task coordination. It aims to allow human users to coordinate casual tasks on the basis of their occasional preferences.

Human-Robot Interfaces for Commanding Tasks

Various user interfaces have been proposed for operating robots interactively, such as use of a handheld device for the remote control of a mobile robot [6] and a multi-touch display for simultaneous control of multiple robots [7]. These interfaces allow real-time control and are suitable for tele-existence, military, and gaming applications in which the user needs to continuously monitor what the robot is doing.

Home robots are designed to complete household tasks by themselves and free the user from spending time on them. Therefore, it is desired that user interfaces for such robots can specify the details of repetitive and/or time-consuming tasks in intuitive ways. For instance, Cooky [12] allows the user to command multiple robots to cooperatively cook meals. Magic Cards [13] allows the user to put cards in a room where the user wants to particular tasks done such as "clean here" and "move objects from here" and "to there." Then, the system drives a card collector robot and delegates each task to an appropriate robot, e.g., cleaning task to a cleaner robot and delivering task to a carrier robot. When

there is an error during the task execution, a printer robot prints error messages. This system and our work share the concept of commanding multiple kinds of agents and getting responses from them. The difference is that their system assumes the task details are clear when the user puts cards in the environment, while ours is capable of handling ambiguous tasks.

Instead of creating new interfaces for commanding robots, making use of social media platforms is proposed [9]. The system uses existing tools originally designed for human-human communication, including text messaging on cell phones, video chatting on personal computers, and an online shared calendar. Text messaging and video chatting both involve text-based dialogues and allow interactive task specifications, similar to the discussion board in our work. Though, there are two major differences. First, the prior work focuses on comparing social media to find the characteristics of each medium as a commanding interface. Our work, however, focuses on proposing a novel use of one social medium, a to-do list. The novelty is in its bi-directional commanding; it allows not only commanding *to* a robot but also *from* a robot. Second, in the prior work, each medium supports one-to-one communication between the user and a robot, but our work supports communication involving multiple users and multiple agents.

To-do List Interface for Task Management

A to-do list is one of the standard tools for task management. There are many web-based to-do list services, which often have an option to share the list with others for computer-supported cooperative work (CSCW). Within the research context, the effectiveness of sharing the lists in a distributed environment was discussed [8]. To-do lists have also often been discussed as part of personal information management (PIM) systems [1]. Intelligent assistance for to-do lists was proposed [5] that divides existing tasks into sub tasks, prioritizes tasks, merges similar tasks, and assigns tasks to automated programs on the basis of "common knowledge" collected in advance from volunteers and online corpora. In these cases, details on each task are shared and discussed with others via other conversation media, typically e-mail. In contrast, our interface has a discussion board that enables seamless integration of information sharing and discussion between human and agents.

Towel [4] is one of the closest examples in that it combines a to-do list interface with a text-based chat window for discussion between the user and a software personal assistant. While Towel provides a chat window for one-to-one human-agent communication to delegate digital tasks to a specific agent, our interface is designed to be a platform for collaborative completion of both physical and digital tasks. For instance, while Towel's chat log is only visible to the user and agent, our discussion board is accessible by all members of the to-do list, including human and agents. This allows support for more complex use cases, e.g., multiple agents propose handling the same task in parallel.

INTERACTION DESIGN

The main interface of the Sharedo platform looks like a standard web-based to-do list service, as shown in Figure 1 and Figure 2. The user can have more than one to-do list and switch between them in accordance with their purpose and context. For instance, one to-do list is for private use shared with the media content agent, while another is for sharing with family users and the cleaning agent, etc. Within each to-do list, multiple tasks can be registered. For each task, its own page shows the task details and a discussion board where human and agents can discuss the task. To identify each user and agent, we use the authorization API of an existing social medium, Twitter. Twitter is also used to send notifications to human users through its private messaging feature called "direct message."

Despite these standard features, the novelty of this interface comes from the platform being capable of hosting not only human-human but also human-agent interaction. In the following subsections, the human-agent interaction enabled on the platform is explained with concrete example scenarios with three agents: the house cleaning agent (@sharerom), the shopping agent (@sharezon), and the media content agent (@sharecnt). These agents are carefully designed to have different characteristics as shown in Table 1 to reveal the use cases of the platform.

Share Tasks

The user can add a task to a to-do list by specifying the title, e.g. "Buy milk," detailed description (optional), assignment, and deadline (optional). Options for the assignment are the user him/herself, a specific member among humans and agents sharing the to-do list, and anyone sharing the list. When a task is added to a list, it is immediately notified to all users and agents sharing the list. Each agent checks if it has anything to do with the task. For instance, the shopping agent is capable of buying milk, but the house cleaning and media content agents are not. In this case, the shopping agent proceeds to the next step of discussion, and the others keep silent.

An agent can also add a task to the list if it needs the help of human users or has some recommendations (Figure 3). For instance, the house cleaning agent might fail to complete cleaning because of errors such as low battery and being trapped in tangled cables on the floor. When the errors can

Agents and their features	**Home Cleaning**	**Shopping**	**Media Content**
Share – add user tasks	✓	-	✓
Discuss – clarify details	✓	✓	-
Discuss – notify progress	✓	-	✓
Complete tasks by itself	✓	✓	-
Tasks are…	Physical	Both	Digital

Table 1. Variety of agents with different characteristics

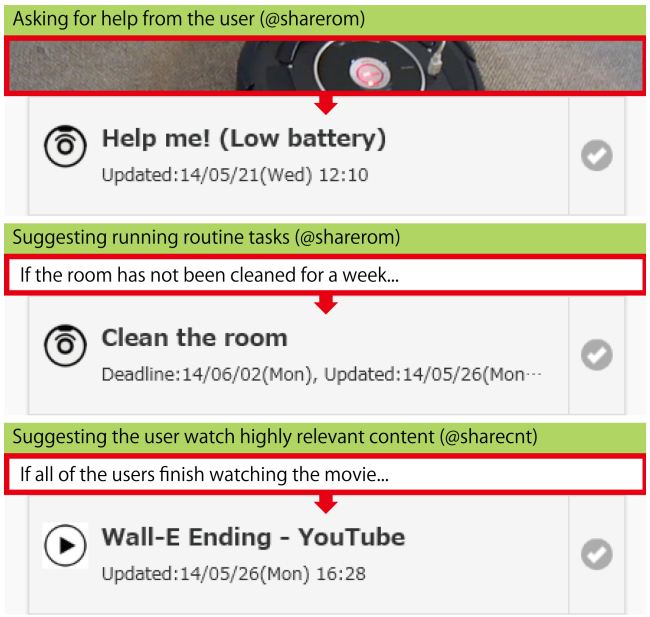

Figure 3. Examples of tasks created by agents

only be fixed by a human, the agent adds the to-do "Help me!" to the list and asks for help. The media content agent might also add a task for human users to watch a video that is considered highly relevant to recently watched videos. Please note that the tasks added to the list can be freely edited or removed if human users think they are not desirable. Once they are removed, the agents stop handling them. Every change the user makes on the Sharedo platform tells something to the agents, which in turn achieves a sort of non-verbal communication.

Discuss Tasks

After a task is shared in a to-do list, the discussion phase starts and continues till it is completed or removed from the list (Figure 2). The discussion takes place on a discussion board provided separately for each task in the list. This allows focused and effective discussion towards task completion.

One major role of the discussion board is to remove any ambiguity in the task and to create a complete task definition required to complete the task. The task definition in general consists of *who* should complete the task and *when* and *how* it should be completed. While *who* refers to one of the humans and agents sharing the list and *when* refers to a specific time and date, what kind of information *how* consists of is different from one task to another. On the Sharedo platform, each agent relevant to the task tries removing ambiguity in its own way in parallel. In other words, each agent has its own task definition and asks the user to fill in blank fields by concurrently posting comments. Such concurrent posts make the discussion board look similar to existing social media such as Twitter.

For instance, adding the task "Buy a broomstick and clean the room" makes both the shopping and house cleaning agents post comments to the discussion board. The shopping agent searches for broomsticks on shopping sites, lists relevant products, and asks the user for a response. The house cleaning agent does not know if the user is going to use the broomstick to clean the room by him or herself, so it searches for time slots when the user is out of the house and offers those slots in which the Roomba robot will clean the room. The user can read these posts and assign the task to one of the two agents to mute further posts from the other agent. The user can even ignore these posts and go to the next step of completing the task by him or herself, making both agents stop posting to the discussion board.

Another example of ambiguous tasks is "Watch the Disney movie 'WALL-E'" accompanied with a YouTube URL (Figure 2). The shopping agent can buy and deliver the Blu-ray disc, shown as example interaction (B), but the media content agent can also show it online, shown as example interaction (A). The media content agent posts an inline YouTube video player to the discussion board.

The other role of the discussion board is to remind the user of the task. Since posting comments on the board triggers the sending of notifications to all users sharing the to-do list, agents sometimes make use of the board to make sure that human users complete the task. This is critical when a deadline is set to a task.

For example, suppose one user adds the task "See video before next meeting" to a to-do list shared with colleagues. The task is accompanied with a YouTube URL in its description and with a specific deadline. The user thinks that watching the video is mandatory to prepare for the meeting but is not sure if all of the colleagues will watch it. When the to-do list is shared with the media content agent, the agent addresses this concern. It posts a comment with an inline YouTube player allowing the colleagues to easily watch the video. Furthermore, it tracks the activity of the colleagues on the comment. When one starts playing the video with the inline player or clicks the link to watch it on the full YouTube site, the agent internally marks the user as "done." The creator of the task can post a new comment such as "Who hasn't watched the video yet?" to make the agent respond with a comment listing the colleagues whose activity has not been observed yet. Since the agent provides clear information on the progress, the creator of the task can be relieved from communicating with each colleague just to make sure whether they have watched the video.

Complete Tasks

Once any ambiguity in a task is removed, the last step is to complete it, mark the task as "done" on the system, and notify all users and agents of the task completion.

The user is allowed to complete tasks at any time, even if the discussion is in progress or if the user has already assigned the agent to handle it. For instance, the user might notice he can "buy milk" on his way back home. There are cases where agents cannot understand subtle nuance in the task

definitions. For example, the user could find time to "clean the room" and actually want to do it by himself rather than asking the house cleaning agent, which does not consider the specific space he is temporarily using for an intricate jigsaw puzzle. Once the user completes a task and marks it as completed on the system, the agents do their best to cancel further actions. For instance, when the cleaning task is marked as done, the house cleaning robot cancels its scheduled cleaning.

Each agent has its own implementation for completing tasks. The task completion procedure can be self-contained as with the case of the house cleaning agent and the shopping agent. These agents can potentially complete the tasks without help from human users. Meanwhile, the procedure can depend on the human users' actions as with the case of the media content agent. It essentially observes the action of each user and marks the task as done when all conditions are met. During this procedure, new tasks are potentially made by agents. For such examples, see the second paragraph in the subsection "Share Tasks."

IMPLEMENTATION

Sharedo Platform

The Sharedo platform is implemented as a set of Java Servlets on the Google App Engine (GAE) server (Figure 4). Both the user client and agents call the Sharedo Web API to communicate with the platform. The agents also have a web hook mechanism, a Web API called by the platform for event notification. The implementation of the user client and agents is separated so that the user has control over the privacy policy (which agent to share what information) and to ensure capability for implementing more agents on the platform. Currently, the house cleaning agent, the shopping agent, and the media content agent are implemented on top of the platform.

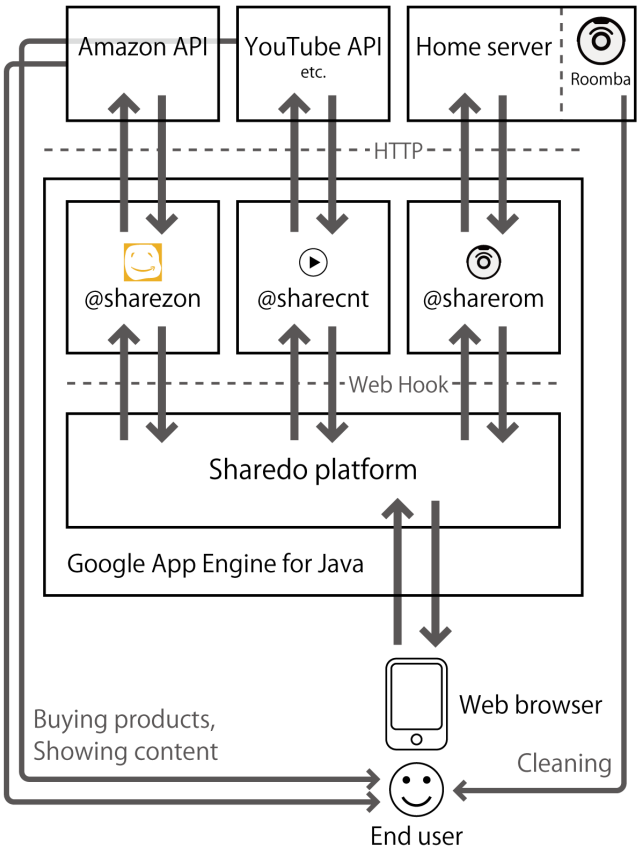

Figure 4. Sharedo platform with multiple agents

User interface – The user interface is implemented with HTML5, CSS3, and JavaScript. All communication between the user interface and the platform is done via Asynchronous JavaScript + XML (Ajax), except for the authentication process that uses the OAuth API provided by Twitter and the initial load of the top page. Ajax allows smooth transitions between pages and provides a better browsing experience.

Event notification – The Sharedo platform notifies users of an event that has occurred in the to-do lists through a direct messaging API provided by Twitter. The notification is sent to their mobile phones or mailboxes, depending on their preferences. The platform also notifies the agents of the event through WebHook, which is a technique for sending HTTP requests to the Web APIs of agents. The current implementation notifies users and agents of events including the start and end of sharing, that is, the insertion, edit, and removal of a to-do, and the posting of a comment.

House Cleaning Agent

The house cleaning agent consists of two major components – the server-side implementation responsible for virtual embodiment of the agent and the client-side implementation responsible for physical embodiment of the agent.

The server-side implementation is installed on the GAE server. The implementation receives an event and responds to it on the Sharedo platform. It has various functions, e.g., to read the details of a task in the to-do list, to respond to comments, and to send instructions to the client-side implementation. When a new task is created, it finds a keyword related to cleaning in the text-based title and a description of the task by matching regular expressions. Then, with permission to access the user's online calendar (Google Calendar) granted beforehand, it looks for a time slot when the user is out of the house and offers a time in which the robot will work. Next, the agent waits for the user's response. If the user approves the offer by posting comments that contain "yes" or other similar words, the client-side implementation is commanded to handle the task at the selected time. If the user rejects the offer, the agent finds another appropriate time candidate. If the user does not respond, the agent will not work when the selected time comes. It will offer a new time for cleaning after the planned time has passed. The time can also be explicitly specified by commenting on it. The number of times for a thorough cleaning job can also be specified; for instance, commenting "twice" makes the robot repeat the cleaning twice. If the users have not added a cleaning task to the to-do list for more

than a predefined period (a week by default), the agent adds the task "Clean the room" in the user's to-do list as a reminder.

The client-side implementation is installed on a personal computer in the user's house. It receives a HTTP request from the server-side implementation to control an iRobot Roomba doing an actual cleaning task in the real world. The Roomba robot is controlled wirelessly with the Roomba Open Interface protocol via a Bluetooth connection. When the task is finished, it notifies the server-side implementation to mark the cleaning task as completed. Otherwise, when a fatal error is reported by the robot that prevents task completion, it notifies the server-side implementation to add a new task that asks a human user for help.

Shopping Agent

While the shopping agent is implemented without any physical embodiment, it certainly affects the real world upon task completion (the product is delivered to the user's house). It is installed on the GAE server that orders products from an online shopping service (Amazon.com). In the future, it might be possible that we substitute the shopping service with a mechanical robot capable of buying and delivering the products to the mailbox.

The implementation finds a keyword in the title and description of the newly posted/edited task in a similar way to the house cleaning agent. It matches regular expressions to extract the keyword from a task, such as "buy something," and then searches for products by using the Product Advertising API by Amazon.com to show the most likely products (up to three) in the discussion board. The user can refine the result by posting a comment with the format "refine keywords" or instruct the agent to search again with "search new keywords." In addition, if the user wants to buy groceries, the user can narrow the results by including the category keyword "#Grocery." The search is repeated until the user is satisfied with the results. When the user approves an offer by clicking the product image on the discussion board, the shopping agent posts a comment with a link to make an order. Finally, when the order has been made, the task is marked as "completed." After the payment process, the ordered item will be delivered to the user.

Media Content Agent

The media content agent is installed on the GAE server and collects information about the URL mentioned in the task description. It posts a message that helps the human users to understand the media content pointed by the URL before they access it. In addition, by substituting the original URL with a redirection URL, the agent tracks which user has accessed the URL.

While the message depends on the content of the URL, a common procedure for the agent is to access the URL and try parsing the contents as a HTML document and retrieve the title. The message always starts with a hyperlink to the redirection URL such as "Please see <title of the web page>."

If the original URL is a YouTube video, the agent posts an inline player. The title text is shown as a hypertext link that navigates to a redirection URL, allowing the agent to track user activity with link clicking. Thanks to the YouTube Iframe API, the agent can also track user activity with the inline video player. When the user clicks the link or plays the video, the agent silently marks the user as "done." When all human users who are supposed to watch the video are marked as "done," the agent posts a congratulation to reward the users and to notify the task creator of it. In addition, the agent searches for highly relevant videos and posts them as a new task to the to-do list. Tasks posted by the agent can be easily distinguished on the list since all tasks made by other users and agents are shown with their social media icons.

Otherwise, when the original URL is not a YouTube video, the agent posts a thumbnail image showing the overview of the website instead of the inline player. While the video-specific interaction is omitted, all the rest remains the same; for instance, when all the users access the redirection URL, the agent congratulates them by posting a message.

PRELIMINARY USER STUDY

Method

We conducted a preliminary user study for two months by recruiting four participants in two families (one family with a 25-year-old male and 27-year-old female and the other with a 36-year-old male and 26-year-old female.) The study aimed to compare use of the Sharedo interface against the direct commanding of agents and to observe how the interface is used in a real setting. Besides the Sharedo interface, the participants were allowed to command an iRobot Roomba robot, access the Amazon website, or see other websites without the help of agents at any time during the user study.

Results

After two months, the participants made 179 to-dos with 111 comments including 61 from agents in 15 to-do lists. The Sharedo interface was used well to perform collaborative work between the participants and the agents.

For several times, the participants pushed the physical button of the Roomba robot to get the room cleaned immediately. Otherwise, they used our interface effectively to postpone their decision and later handled the cleaning on their own, asked their partner to do it, or made the robot do it at the correct timing. While the cleaning tasks done by the robot usually succeeded, one participant reported, *"I worried about the robot even when I didn't get messages while I was out."* and *"I came to feel more like cleaning the room by myself when possible. Otherwise, I often had to arrange the room to get it ready for robot cleaning."* Another participant said, *"When I made the robot clean the room, I wanted to specify the area to be cleaned. I did not want my Roomba to run over my belongings on the floor, though there was no other option than moving them by myself. Furthermore, I*

think such an option should not be told to other users and agents through text-based communication."

Discussions about the shopping agent were interesting. Two of the participants were a married couple who reported, *"We felt like the shopping agent was a friend. The fact that it has its Twitter account as we do and it communicates with us through the account makes it more familiar. The Amazon.com website also has recommendations for us, but the feeling was very different,"* and, *"The recommendations made by the agents were sometimes beyond our expectations, expanding our choice."*

DISCUSSION AND FUTURE WORK

According to the preliminary user study, the platform itself was proved to work stably and is thought to be extendible enough to see a greater variety of agents being implemented in the future. The process of defining task details through the discussion board interface got favorable feedback from the users. While the board could not be used to get an immediate response from the Roomba robot, it has the potential to help create long-term good relationship between the user and the robot.

The server-side implementation of agents could be more convenient for the user if they can utilize the history of completed tasks. Agents could learn the timing patterns for cleaning a room and remember favorite products for shopping with the help of programming-by-example techniques. Some users complained that the text-based communication on the discussion board for specifying the details of cleaning tasks is not thorough enough. To address this issue, our future work will improve the agent implementation to integrate more interactive user interfaces within the discussion board.

Regarding the users' preference on the relationship with their agents through Twitter accounts, our future work might include more integration of our interface with Twitter. While the current implementation only uses a direct messaging API so that communication between the user and the agent cannot be seen from other followers of the user on Twitter, it might be interesting to see what happens when we allow public communication between them, which is shown in the Twitter timeline and visible to anyone on the Internet.

Collected comments from the study suggest that the platform can enhance communication between people. Deploying the interface as a communication medium for the user and professional housekeepers instead of the cleaning agent would also produce interesting results.

CONCLUSION

We presented a user interface design with which a user can share, discuss, and complete to-dos together with people and agents. Its implementation, named "Sharedo," can be accessed at http://www.sharedo.info/ and comes with three agents: house cleaning, shopping, and media content agents. A preliminary study showed that the platform could help remove ambiguity in task definition through the standard workflow of task management, showing a promising way to coordinate tasks for human-agent cooperation.

ACKNOWLEDGMENTS

This work was supported in part by JST, CREST.

REFERENCES

1. Bellotti, V., Dalal, B., Good, N., Flynn, P., Bobrow, D. G., and Ducheneaut, N. What a to-do: studies of task management towards the design of a personal task list manager. In *Proc. of CHI '04*, 735-742.
2. Campbell, A., and Wu, A. Multi-agent Role Allocation: Issues, Approaches, and Multiple Perspectives. *Autonomous Agents and Multi-Agent Systems 22(2)*, Springer (2011), 317-355.
3. Chalupsky, H., Gil, Y., Knoblock, C., Lerman, K., Oh, J., Pynadath, D., Russ, and T., Tambe, M. Electric Elves: Applying Agent Technology to Support Human Organizations. In *Proc. of AAAI IAAI '01*, 51-58.
4. Conley, K. and Carpenter, J. Towel: Towards and Intelligent To-Do List, In *Proc. of AAAI Spring Assistants*.
5. Gil, Y., Ratnakar, V., Chklovski, T., Groth, P., and Vrandecic, D. Capturing Common Knowledge about Tasks: Intelligent Assistance for To-Do Lists, *ACM Transactions on Interactive Intelligent Systems 2(3)*, Article 15.
6. Fong, T., Thorpe, C., and Glass, B. PdaDriver: A Handheld System for Remote Driving. In *Proc. of ICAR '03*.
7. Kato, J., Sakamoto, D., Inami, M., Igarashi, T. Multi-touch interface for Controlling Multiple Mobile Robots. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems (CHI EA '09)*, 3443-3448.
8. Kreifelts, T., Hinrichs, E., and Woetzel, G. Sharing to-do lists with a distributed task manager. In *Proc. of ECSCW '93*, Kluwer Academic Publishers (1993), 31-46.
9. Ma, X., Yang, X., Zhao, S., Fu, C., Lan, Z., and Pu, Y. Robots in my Contact List: Using Social Media Platforms for Human-Robot Interaction in Domestic Environment. In *Proc. of APCHI '12*, 133-140.
10. RememberTheMilk. http://www.rememberthemilk.com/
11. Scerri, P., Pynadath, D., Johnson, L., Rosenbloom, P., Si, M., Schurr, N., and Tambe, M. A Prototype Infrastructure for Distributed Robot-Agent-Person Teams. In *Proc. of AAMAS '03*, 433-440.
12. Sugiura, Y., Sakamoto, D., Withana, A.I., Inami, M., and Igarashi, T. Cooking with Robots: Designing a Household System Working in Open Environments. In *Proc. of CHI '10*, 2427-2430.
13. Zhao, S., Nakamura, K., Ishii, K., and Igarashi, T. Magic Cards: A Paper Tag Interface for Implicit Robot Control. In *Proc. of CHI '09*, 173-182.