

Sharedo: To-do リストによる人-ロボット間のタスク共有

Sharedo: To-do List for Sharing Tasks between Human and Robots

加藤 淳 坂本 大介 五十嵐 健夫*

Summary. ユーザ間で To-do 共有が行える Web サービスにおいて、ロボットにユーザと同等の権限を与えて、ユーザとロボットの間でタスクの分業を行えるようにした試作システム Sharedo について報告する。本システムには、リスト単位で共有相手を選べる機能がっている。また、To-do ごとにコメント欄があり、曖昧な点について共有相手と対話できるようになっている。

1 はじめに

ロボット技術が発達するにつれ、様々な場でロボットの活躍が期待されるようになってきた。一般家庭向けには自動掃除機¹が販売され、研究段階では、ロボットが調理をする [5] アプリケーションなどが提案されている。このような実用目的のアプリケーションでは、システムがユーザの意図を汲むためのインタラクション設計が重要である。既存研究では、ユーザがロボットを直接操作でき、明示的な指示を出せるジョイスティックなどのユーザインタフェースのほか、実際の部屋の中に掃除などの操作内容を記したカードを置いたりする [7] タスク指向のユーザインタフェースが提案されている。

これらの提案ではロボットがこなすべきタスクの全体像は明確であり、そのやり方を指示する手法が研究対象となっている。しかしながら、実際の生活にロボットが入ってきたとき、ユーザがロボットにできることの全てを常にロボット任せにするとは限らない。例えば、忙しいときはロボットに掃除を頼むが、時間に余裕のあるときは自分で掃除をしたいというニーズが考えられる。したがって、人とロボットでどのようにタスクを分業するかという一段メタな問題も含めたインタラクション設計が必要となる。

本稿では、ユーザがやるべきことを記入する To-do リストをロボットと共有し、人とロボットの分業体制を実現できるクラウドベースの Web アプリケーション「Sharedo」を提案する。Sharedo では、複数のユーザとロボットが To-do リストを共有する。リストの共有メンバーは、リストに To-do 項目を追加、編集、削除したり、自身が To-do をこなした場合に状態を完了に変えたりできる。また、To-do ごとにコメントを残せる機能があり、To-do の内容について曖昧なところがあれば共有メンバー間で議論できるよ

うになっている。本研究で提案するインタフェースでは、ユーザは、通常の To-do リストと同様に To-do を書き込み、それを「誰か」と共有して一緒に To-do の完了を目指す。「誰か」とは実際に To-do を行える主体であるが、これが人でもロボットでもよく、その別を問わない点において、Sharedo は Computer-Supported Cooperative Work(CSCW) をロボット向けに応用した新しい事例であると考えられる。

2 関連研究

2.1 ロボットの同期的な操作のための UI

ロボットとの対話技術についての研究は様々にあるが、人がロボットを同期的に操作するための技術は最も古くから研究されてきた。例えば、軍事ロボットのためのインタフェース [2] がある。Tele-existence [6] ロボットや、人を模したロボットのためのインタフェース [4] などよく知られている。これらのインタフェースでは、ロボットが動作する間、ユーザが継続的に操作を行う必要がある。しかし、一般的なユーザが家庭用ロボットに寄せる大きな期待の一つは、ロボットが家事等の作業をユーザの代わりにこなしてくれて、ユーザがその作業から物理的・時間的に解放されることである。この点においては、同期的な操作を伴うロボットは一般家庭に受け入れにくい存在であると考えられる。

2.2 ロボットの非同期的な操作のための UI

電化製品は、ユーザを家事から物理的・時間的に解放してくれる。例えば全自動洗濯機は、ユーザから洗濯の方法を指示された後、全自動でその作業を終えることができる。その間ユーザは他の家事をしたり、家族との時間を過ごしたりできる。このように、すでに販売されている自動掃除機などの家庭用ロボットも含め、生活に近い機械では非同期性が重要である。これまでも、紙のカードをユーザが環境に配置する手法 [7] のように家庭用ロボットとの非同期的な対話手法に関する研究が行われてきている。Sharedo も非同期的な作業指示を行えるものだ

Copyright is held by the author(s).

* Jun Kato, 東京大学 / 日本学術振興会特別研究員 DC, Daisuke Sakamoto and Takeo Igarashi, 東京大学 / JST ERATO 五十嵐デザインインタフェースプロジェクト

¹ iRobot Roomba. <http://www.irobot.com/>

が、指示された作業を行う主体が人でもロボットでもよい点において、先行研究よりもロボットと人が対等な立場で接するインタラクションを実現しようとするものである。

2.3 To-do リストによるタスク管理

本研究で提案するインタフェースは、個々人がやるべきタスクを一覧にした、いわゆる To-do リストを応用したものである。近年では、Web ベースでユーザがどこからでも利用できる様々な商用サービス²が提供されている。研究では Personal Information Management (PIM) に関連して取り上げられることが多い [1]。また、複数人で目標を共有して作業にあたるインタフェースの提案は CSCW で数多くあり、To-do リストを分散環境で共有する提案 [3] もなされている。本研究は、このように人々の間でタスク管理によく使われる To-do リストをロボットと共有する試みである。

3 Sharedo でのインタラクション

Sharedo ではユーザとロボットが同列に扱われ、個々に Twitter³ アカウントを持っている。本節では、Sharedo 全般のインタラクションについて述べた後、現在実装されている掃除ロボットと買い物ロボットに固有のインタラクションについて詳述する。

3.1 To-do リストの共有

ユーザは、まず図 1(a) のように Twitter アカウントを用いて Sharedo にログインする。次に、他の人が作った To-do リストのトップページにアクセスして、当該リストの共有をリクエストする。相手が共有リクエストを承認すると、各メンバーが図 1(b) のような To-do 一覧にアクセスできるようになる。相手がロボットの場合、共有リクエストは即座に承認される。こうして、複数のユーザとロボットが To-do リストを共有でき、以降のインタラクションの準備が整う。

ユーザは、不要と考えるロボットサービスとはリストを共有しなければよい。リストごとに共有するロボットを変えて、複数のリストを使い分けることもできる。また、リストの作成者はいつでもリストの共有メンバーを削除できる。したがって、ロボットサービスが不要になった際は当該アカウントとのリストの共有を停止して、ロボットのユーザ情報へのアクセス権を剥奪できる。

3.2 To-do の追加

ユーザは、図 1(c) のような画面で掃除や洗濯、買い物などの To-do をリストに追加する。この際、実行者を「自分」「特定の相手」「誰でもよい」の三通りから選ぶ。To-do は図 1(d) のような個別ページを持ち、必要に応じて編集・削除できる。また、To-do には「完了」「未完了」を表す状態フラグがあり、To-do リストを共有しているメンバーが誰でも状態を変えられるようになっている。

ロボットが To-do を追加することもある。例えば、しばらく掃除タスクが実行されていない場合に掃除ロボットが「掃除する」To-do を追加してユーザをリマインドすることがある。この場合、実行者は「誰でもよい」ことになっており、ユーザは、タスクが不要なら To-do を削除すればよい。

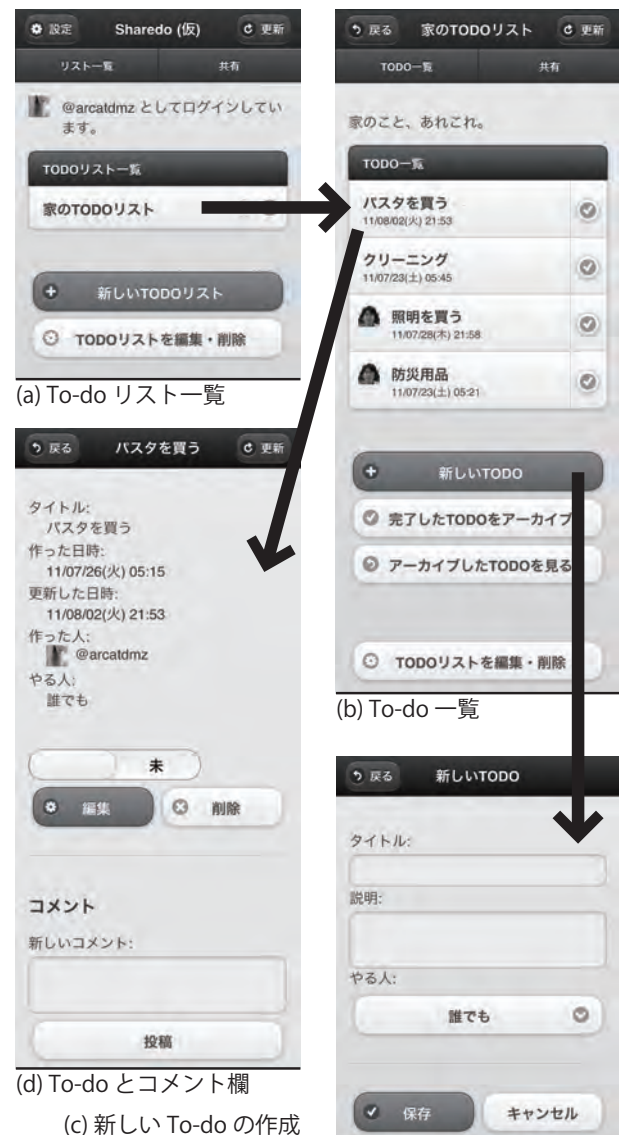


図 1. Sharedo の主な画面

² Remember The Milk. <http://www.rememberthemilk.com/>

³ Twitter. <http://twitter.com/>

3.3 コメント欄での対話

To-do は、To-do リストに追加された短いテキストだけでは具体性に欠け、追加した本人にしか実行しようがないことも多い。既存研究ではメールなど他メディアとの連携によって To-do に関する議論を深める方法が提案されているが、本提案では To-do を表示するインタフェース自体にコメント欄を設置し、ユーザとロボットが対話を通じて To-do に関する曖昧性を除去できるようにしている。

ロボットは、自身が共有している To-do リストで「未完了」の To-do が追加・編集された際、「特定の相手」で明示的に指名されるか「誰でもよい」場合に、To-do がロボットにこなせる内容か判断する。不可能な場合は無視するが、こなせる可能性があれば実行に必要な情報を収集する。そして、曖昧性が残ったらその旨をコメントし、ユーザの返答を待つ。

3.4 To-do の実行

ユーザは、好きなタイミングで To-do を実行して状態を「完了」に変えられるが、実行に際して To-do の実行者を「自分」に設定し、他のメンバーと作業が被らないように明示することが望ましい。

ロボットは、タスクを実行可能な状態になったら、ユーザの最終的な確認の後、適切なタイミングでタスクを実行しようとする。無事タスクを完了できることが確定したら、To-do の状態が「完了」に書き換えられる。もしタスクの実行中にエラーが起きたら、コメント欄に詳細が書き込まれる。さらに、バッテリー残量低下など、ロボット単体では回復不能だが人の助けを借りれば回復可能なエラーが起きた場合は、その内容の To-do が追加される。

3.5 掃除ロボット (@sharerom)

@sharerom は家を掃除してくれるロボットを表すアカウントである。ユーザが「掃除する」というキーワードを含む To-do を作成すると、@sharerom はユーザが家にいない時間帯を調べ、その時間に掃除を試みる旨をコメント欄に書き込む。ユーザはコメント欄で「はい」などと返信して提案を了承し、実際に掃除させることができる。時間帯が不服の場合、「いいえ」と返信して次候補を出させたり、「4日午前10時」のように明示的に時間を指示したりできる。また、「2回」などと返信すると、指定された回数分、念入りに掃除をさせることが可能である。このようにロボットが解釈できる返信をした場合、ロボットは、確認のために掃除の予定を返信してくる。ユーザが了承の返事をしなければ、ロボットは予定時刻を過ぎても掃除を行わず、新たな予定を提案してくる。

また、掃除する To-do が過去1週間作成されていない場合、ロボットは「掃除する」という To-do を作成してユーザに掃除を促す。

3.6 買い物ロボット (@sharezon)

@sharezon は買い物をして家に届けてくれるロボットを表すアカウントである。ユーザが「～を買う」という To-do を作成すると、「～」部分をクエリとして商品データベースに問い合わせ、最もそれらしい順に上位3品目までの商品紹介をコメント欄に書き込んでくる。ユーザは「絞り込み」や「再検索」にキーワードを続けて返信することで、絞り込み検索や再検索を指示できる。また、食品なら「#Grocery」など商品カテゴリを表すワードを含めることで効率的に検索できるようになっている。

ユーザは、ロボットが提示する商品の中に目的のものがあれば、商品名をクリックしてロボットに購入希望であることを伝える。ロボットは、購入の準備ができると、ユーザに決済ページへのリンクを提示して決済を促す。決済後、商品が家に届く。一度決済した商品については、同じキーワードを指定した際に優先的に提示されるようになる。

4 実装

本提案の実装は、クライアント側ユーザインタフェースを提供するプラットフォームと、To-do リストを共有してユーザとコメントを介したインタラクシ

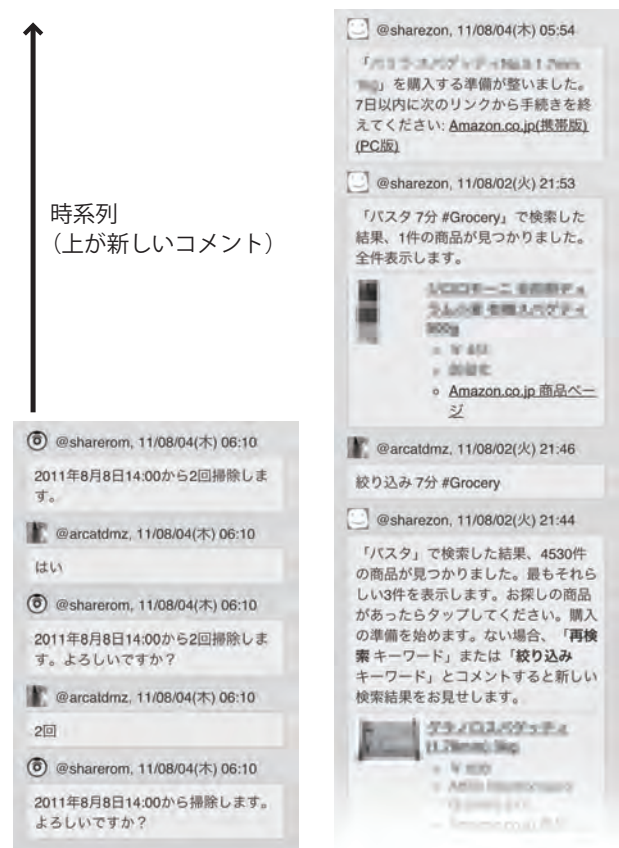


図 2. @sharerom, @sharezon とユーザの対話例

ンを行うロボット群，そして各ロボットと対になったロボット操作プログラムに大別される(図3)．現状，プラットフォームとロボットは共に Google App Engine (GAE) 上に実装されている．それぞれ HTTP リクエストを受け取って処理を行う Web API を持ち，お互いの API を呼び出す Web Hook で通信を行う．そして，ロボットがロボット操作プログラムに指示を出すことで，ユーザがロボットによるサービスを楽しむことができるようになっている．プラットフォームとロボット群の実装を分けたことで，ロボット及びロボットが知ることのできるユーザ情報をプラットフォーム側で制限でき，ユーザのプライバシーが確保される．また，将来的に多種のロボットがユーザと To-do リストを共有してサービスを提供できる拡張性が担保されている．

以降，プラットフォームと各ロボット(ロボット)の実装について詳述する．

4.1 Sharedo (プラットフォーム)

Sharedo プラットフォームは GAE サーバ上に Java の Servlet として実装されている．

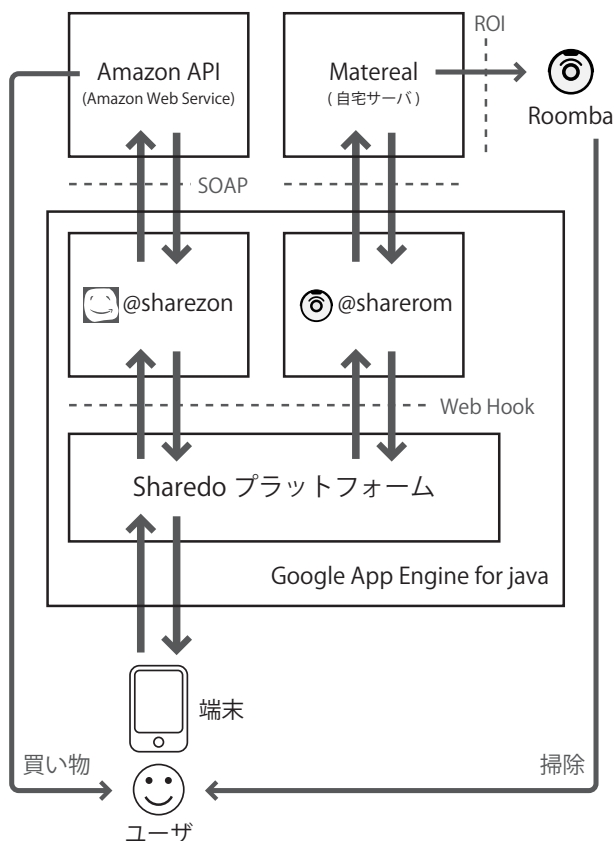


図 3. Sharedo の実装

4.1.1 ユーザ認証とプッシュ通知

Sharedo は Web ベースのアプリケーションであり，単体ではユーザに対してイベントをプッシュ通知するための手段を持たない．To-do 管理のアプリケーションを提案する既存研究では，メールや専用クライアントアプリケーションをプッシュ通知に利用する例が多い．一方，本研究では，日常生活のなかでつぶやきを通じてコミュニケーションを取るマイクロブログの利用が一般的となった状況を鑑みて，マイクロブログサービス Twitter の Direct Message (DM) というメール相当の仕組みを用いてプッシュ通知を実現した．同時にユーザ認証も Twitter を利用して行い，メールアドレスではなく Twitter アカウントでユーザを識別することとした．これによりユーザは，To-do リストの共有相手を追加する際に，メールアドレスより一般的に短く覚えやすい Twitter アカウントを入力するだけで済むようになっている．なお，本節で述べた目的に叶うサービスプロバイダや実装手法を用いれば，本提案は Twitter に依存しないように実装可能である．

4.1.2 Web API

Sharedo プラットフォームはクライアントとロボット向けに Web API を提供している．Web API は，クライアントのユーザ認証やロボットの Web Hook 呼び出しの際にプラットフォームが発行する UUID (汎用一意識別子) をリクエストに含めることで，適切な権限の下で実行できる．例えば Cookie に UUID がセットされた状態で `http://www.sharedo.info/lists/json` にアクセスすると，UUID に対応するユーザが共有している To-do リストの一覧が JSON 形式で返ってくる．

4.1.3 ユーザインタフェース

クライアント側ユーザインタフェースは HTML5，CSS3 および JavaScript で構築している．サーバとの通信は，認証画面および認証後のトップページの読み込みを除き全て Ajax (JavaScript を用いた非同期通信) によって行われ，ユーザとのインタラクションがページ読み込み時の画面書き換えで阻害されないように配慮している．

4.1.4 Web Hook によるイベント通知

Sharedo プラットフォームは，何らかのイベントが起きると Web Hook を用いてロボットにイベントを通知する．現時点でサポートされているイベントの種類は，To-do リストの共有・共有解除，To-do 項目の作成・編集・削除，To-do 項目へのコメント投稿である．ロボット側では，これらのイベントに応じた処理を書くことが求められる．

4.2 掃除ロボット (@sharerom)

@sharerom のボットの実装は、イベント通知があるたびに、To-do 項目の内容やユーザのコメントを読み込んで、当該 To-do のコメント欄に返信したり、ロボットに対して指示を送ったりするだけの単純なものである。とくに、ユーザの入力を解釈する部分は、自然言語処理などを用いず、正規表現によりパターンマッチのみで実装されている。

ボットには予め、ユーザが利用しているオンライン・カレンダー (Google Calendar) へのアクセス権が与えられている。ボットは、Google Calendar API を利用してユーザに在宅以外の予定が入っている時間帯を取得し、ユーザに対して掃除する時間の案を提案する。

@sharerom のロボットの实装は、自宅サーバ上で動作するロボット操作プログラムとロボット本体に別れている。ロボット操作プログラムはインターネット側から簡易 HTTP サーバとしてアクセスできるようになっており、ボットからの指示は HTTP リクエストとして送られてくる。実装には Java 及び Java 用のロボット操作ツールキット Matereal[8] が使われている。ロボットは iRobot 社 Roomba に Bluetooth モジュールを挿したものを扱い、ロボット操作プログラムとは Roomba Open Interface (ROI) プロトコルを用いて通信を行う。ROI には「掃除を行う」という高レベルな命令が用意されており、ロボット操作プログラムはこれを指定時刻に呼び出す。掃除をし終えた段階でロボット操作プログラムがボットに Web Hook を用いて通知を行い、ボットがタスクの状態を「完了」に書き換える。

4.3 買い物ロボット (@sharezon)

@sharezon の実装は Amazon 社が提供する通信販売を利用しており、機械装置を用いた狭義のロボットとは異なる。しかし、To-do リストを介して買ってくるべき商品を指示すると、適当な時間が経った後に商品がポストや宅配ボックスに入っているという現象をユーザの立場で家の中から観察すれば、これは広義のロボットアプリケーションであると言える。少なくとも、本提案は将来的に狭義のロボットに対して買い物を指示するインタフェースとしても用いることができるため、インタラクション研究の観点からすれば、実装部分が機械であろうとなかろうと議論の一般性は損なわれまいと考える。

ボット部分の実装は @sharerom と同様に比較的単純である。ユーザの登録した To-do 項目のタイトルやコメント欄からキーワードを取得し、Amazon 社が提供する Product Advertising API を用いて商品検索を行い、結果を 3 件までコメント欄に投稿する。ユーザが返信を続ける限り検索をやり直し、ユーザが満足して商品リンクをクリックしたら、同 API を用いて当該商品の決済ページへのリンクを作

成して提示する。そして、ユーザが決済ページを訪れるとタスクの状態が「完了」に書き換えられる。

5 実運用と考察

我々は、Sharedo⁴を約 2ヶ月間 2グループ 4名に自由に使ってもらった。一方のグループは @sharezon および @sharerom と、もう一方のグループは @sharezon と To-do リストを共有した。全体として、15 個の To-do リストに対して 179 項目の To-do が作成され、ロボットからの 61 個を含む 111 個のコメントが書き込まれた。いずれのロボットも割り振られたタスクを正常にこなすことができたため、ロボットが To-do の作成やコメントの書き込みを通じてエラーを通知する機能は使われる機会がなかった。本節ではこの実運用の過程で得られた知見を述べ、将来の課題を明らかにする。

5.1 オンラインカレンダーとの連携

本システムの運用中、ユーザから To-do が完了した履歴を一覧したいという要望を受けた。例えば、掃除する To-do の履歴を見ればいつ部屋を掃除したか分かる。我々は、To-do の完了履歴だけでなく、ロボットの稼働予定も含めたスケジュールの情報をユーザが普段使っているオンラインカレンダーで閲覧できるようにしてユーザの利便性向上を図る予定である。

5.2 コメント欄とコマンドラインの比較

本提案では、ユーザはコメント欄を介してロボットに対する細かい指示を行う。ユーザがテキストベースで入力を行う点はコマンドラインインタフェースと同様であり、ユーザが慣れれば短いやり取りで詳細な指示出しができる。ただし、プログラムの出力にテキストでなくハイパーテキストを使うことで、例えば買い物ロボットの提示する商品をクリックするだけで選択の意図をロボット側に伝えられるなど、コマンドラインより直感的に対話できるようになっている。この出力部分を作り込むことで、個々のロボットに最適化された高度なユーザインタフェースを構築できるだろう。

また、コマンドラインではユーザとプログラムが一对一で対話するが、コメント欄では複数のロボットが同時に反応する可能性がある。例えば「掃除するための用具を買う」という To-do には @sharerom と @sharezon が共に反応する。現状、To-do をどちらか一方に割り当てるか、ロボットのアカウント名を返信に含めることで、一方のロボットのみとやり取りできる。今後、ロボット間の優先順位をユーザが決められるようにするなど、よりきめ細かい対応ができるような拡張が望まれる。

⁴ Sharedo. <http://www.sharedo.info/>

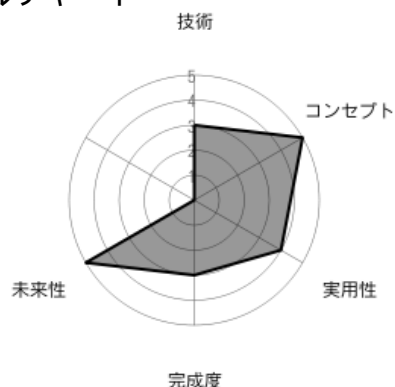
5.3 ロボット用インタフェースの対人応用

本提案のようにロボットがユーザと非同期的なコミュニケーションを取る場合、ロボットの身体性が薄れ、インタラクションの様式は Human-Agent Interaction に近くなる。このため、人工知能におけるチューリング・テストに類似した問題が生ずる。すなわち、タスクを実際に実行した主体が人なのか、それとも高度なロボットなのかという判断が難しくなる。例えば、不在中に部屋を掃除したのがロボットではなく家事代行業者の人でも、買い物をして宅配ボックスに投函したのがロボットではなく通信販売と宅配業者の人でも、ユーザにとって本質的な差はない。裏を返せば、非同期的なロボット向けインタフェースは他人に対する指示出しに応用できる可能性がある。将来的には、本インタフェースを家事代行業者との情報交換に利用し、ロボットの場合との比較研究を行うこともできるだろう。

参考文献

- [1] V. Bellotti, B. Dalal, N. Good, P. Flynn, D. G. Bobrow, and N. Ducheneaut. What a to-do: studies of task management towards the design of a personal task list manager. In *Proc. of CHI '04*, pp. 735–742. ACM, 2004.
- [2] T. Fong, C. Thorpe, and B. Glass. PdaDriver: A Handheld System for Remote Driving. In *Proc. of ICAR '03*. IEEE, 2003.
- [3] T. Kreifelts, E. Hinrichs, and G. Woetzel. Sharing to-do lists with a distributed task manager. In *Proc. of ECSCW'93*, pp. 31–46. Kluwer Academic Publishers, 1993.
- [4] D. Sakamoto, T. Kanda, T. Ono, H. Ishiguro, and N. Hagita. Android as a telecommunication medium with a human-like presence. In *Proc. of HRI '07*, pp. 193–200. ACM, 2007.
- [5] Y. Sugiura, D. Sakamoto, A. Withana, M. Inami, and T. Igarashi. Cooking with robots: designing a household system working in open environments. In *Proc. of CHI '10*, pp. 2427–2430. ACM, 2010.
- [6] S. Tachi. *Telexistence*. World Scientific Publishing Co., Inc., 2009.
- [7] S. Zhao, K. Nakamura, K. Ishii, and T. Igarashi. Magic cards: a paper tag interface for implicit robot control. In *Proc. of CHI '09*, pp. 173–182. ACM, 2009.
- [8] 加藤 淳, 坂本 大介, 五十嵐 健夫. matereal: インタラクティブなロボットアプリケーションのプロトタイピング用ツールキット. In *Proc. of WISS '10*, Dec 2010.

アピールチャート



未来ビジョン

現時点では多くの家庭用ロボットの完成度は人に遠く及ばない。この間を縮める工学的な努力をやめるべきでないというのは論を待たないが、一方で、より至近の未来、実世界において我々が情報技術による恩恵に与るにはどうしたらよいただろうか。以下に、二つの未来ビジョンを提示したい。

まず、我々は、ロボットに全ての雑務を任せ、人は移動式ベッドに寝そべるだけの全自動な未来ではなく、人が得意なことは人が、ロボットが得意なことはロボットが行う「積極的な分業」の未来こそが幸せな解であると考えている。現状でも家電製品がユーザの余暇時間を増やしており、これも分業の一形態ではあるが、人にかかる負担を減らすためだけに機械を用いる「消極的な分業」が多いのではないかと。人が楽しんで創造性を発揮できることをロボットにさせない自由があるべきだ。本提案は、その

ための第一歩として、タスクの実行主体を選ぶ権利をユーザに提供するシステムである。

次に、我々は、人が情報システムの指示通りに実世界タスクをこなすことによって、自分でやり方を考えたときより楽にタスクをこなせる場合が多々あることに着目し、ロボットのように「自動的な人間」として動くことをもっと肯定的に捉えたいと考えている。例えば、本システムは掃除ロボットに回復不能なエラーが起きると人にタスクを割り当て、人をフェイルセーフとして利用することで掃除という大目標を確実に遂行できる。実社会では Operations Research が組織の意思決定に使われるなどマクロな応用例が目立つが、我々は個人単位のタスクに興味がある。例えば、Human Computation は遠隔地の知的労働力を計算能力に変換する手法だが、この実世界版として Human Actuation というものが可能ならどのような形態になるのか考えてみたい。