

チャット形式で対話するネットワーキング秘書ボット「YouBot」

竹形誠司 田中久美子

東京大学大学院情報理工学系研究科

takegata@cl.ci.i.u-tokyo.ac.jp, kumiko@i.u-tokyo.ac.jp

1. はじめに

近年、インスタント・メッセージャーやtwitterのように、短いテキストで情報交換を行うことが一般的に行われるようになってきている。しかし、相手のユーザがログイン中でなければ彼または彼女の様子を知ることはできない。そこで、ユーザがログインしていない場合に、ユーザに代わって問い合わせに応答する秘書ボットがあれば便利である。秘書ボット同士をネットワークで接続することにより、ユーザ同士がスケジュールやその他の情報を共有することができる。そのためには、個人情報や業務上の機密情報など、セキュリティの問題にも配慮する必要がある。本稿ではチャット形式の対話インターフェイスを備え、ネットワーク経由で相互に情報を交換する秘書ボットシステムを提案する。

2. 関連研究

人間の言葉で対話するシステムは、草分け的存在のWeizenbaum[1]や、ローブナー賞を3度受賞したWallace[2]など、数多くの研究がなされてきたが、実用的なシステムは少ない。本稿で提案するシステムは、スケジュールやToDoリストの管理など、一般に電子秘書(またはパーソナルアシスタント)と呼ばれる実用的な機能の実現を目標としている。本稿では電子秘書やパーソナルアシスタントなどのシステムを総称して「秘書ボット」と呼ぶ。

秘書ボットに関する研究には、Nguyenら[3]やCalo Project[4]などがある。これらの研究ではユーザとシステムのインタラクションに主眼が置かれており秘書ボット同士の間の情報の交換については検討されていない。

本提案では、複数の秘書ボットをネットワークで接続し、相互に情報を交換する点に主眼を置いている。秘書ボットは個人情報や業務上の機密を扱うため、情報交換時のセキュリティ確保が重要な課題となる。本稿では、交換情報におけるセキュリティ管理の枠組みについても述べる。

3. 秘書ボットに求められる機能

本システムの目指す秘書ボットは、ユーザに代わってスケジュール管理や情報の収集、関係者との連絡などを行うものである。この秘書ボットに求められる機能は、メインのユーザからの指示に従って情報を管理する機能(対メインユーザ処理)と、他のユーザからの問い合わせに答える機能(対他ユーザ処理)の2つに大きく分けることができる。たとえば、メインユーザが自分のスケジュールを確認するのは前者、他のユーザからメインユーザのスケジュールを尋ねられたときに答えるのは後者の機能である。

他のユーザから問い合わせを受けた場合に、秘書ボットがそれに答えて良いかどうかは、問い合わせの内容や相手のユーザとメインユーザの関係による。本システムでは情報のセキュリティを適切に保つために、秘書ボットが扱う情報にセキュリティ属性、通信を行う相手にセキュリティ権限を付与し、これらを照合する。たとえば、「オフィシャル」のセキュリティ属性が付与された情報に関しては、「オフィシャル」のセキュリティ権限を持つユーザの問い合わせにのみ応答する。

本システムでは他のユーザの秘書ボットへの問い合わせを、秘書ボット同士の通信(ボット間通信)によって行う。図1にボット間通信の概念を示す。

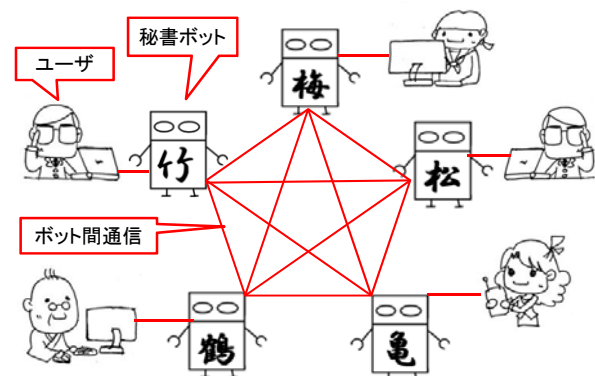


図1: ボット間通信

たとえば、竹下さんというユーザが松井さんというユーザのスケジュールを知りたい場合、竹下さんは自分の秘書ボット(竹ボット)に対して松井さんの予定を尋ねる。すると竹ボットは松井さんのボット(松ボット)との間で情報交換を行い、竹下さんにその情報を伝える。つまり、ユーザは自分のボットとだけ通信を行えばよい。図2に対メインユーザと対他ユーザの処理の流れを示す。

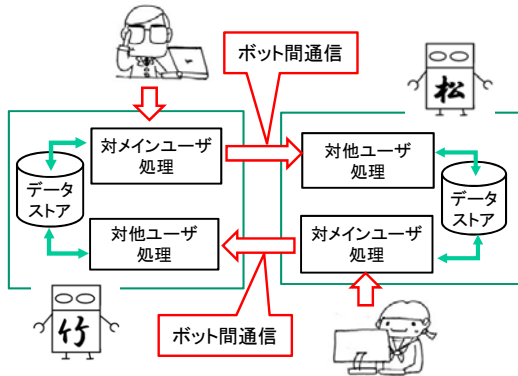


図2: 対メインユーザ処理と対他ユーザ処理

対メインユーザ処理と対他ユーザ処理から抽出した情報は共通のデータ領域に保存する。このデータ領域を本システムではデータストアと呼ぶ。

4. 対メインユーザ処理

4.1 メッセージ処理の流れ

本システムでは、ユーザと秘書ボットの間および秘書ボットと秘書ボットの間両方にチャット通信(インスタントメッセージング。以下IMと表記)のプロトコルであるXMPP¹を利用する。

ユーザはXMPP対応のIMクライアントソフトウェア(Spark²、GoogleTalk³、iChat⁴など)を使って秘書ボットと通信を行う。対メインユーザ処理の概要を図3に示す。

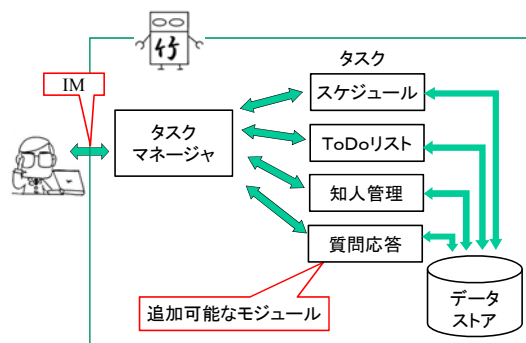


図3: 対メインユーザ処理

秘書ボット内部ではメインユーザとの間のメッセージの受け渡しを「タスクマネージャ」が担当する。タスクマネージャは、メインユーザから受け取ったメッセージを「スケジュール管理」や「ToDoリスト」などのサブルーチンに転送する。これらのサブルーチンは、追加可能なモジュールとして実装される。これらのモジュールを本システムでは「タスク」と呼ぶ。

Nguyenら[3]のシステムではダイアログマネージャと呼ばれる機能ブロックがメッセージの解釈を行い、その結果によって呼び出すタスクを決めている。本システムではタスクマネージャがタスクの呼び出しを行うが、その際にメッセージの解釈を行わず、そのままタスクに転送する。タスクは受け取ったメッセージの解釈を試み、処理可能であれば応答を返し、そうでなければnullを返す。このような方式を取ることで、タスクの独立性を高めることができる。Nguyenら[3]のシステムではタスクを追加するたびにメッセージの解釈ルーチンを修正する必要があるが、本システムではその必要がない。

現状では、あらかじめ指定した順にタスクを呼び出し、最初に応答した(null以外を返した)タスクの回答を採用するが、順位の低いタスクがより適切な回答を持っている場合も考えられる。最適な回答を選択する方法は今後の課題である。

4.2 メッセージからの情報の抽出と保存

メインユーザからのメッセージをどのように解釈し、どのような情報を抽出するかは、タスク側で自由に決めることができる。Weizenbaum[1]やWallace[2]は予め登録したパターンとメッセージを照合している。また、Nguyenら[3]やCalo Project[4]ではBDIモデル[5]を用いている。本システムはどちらにも対応でき、両者を混在させることも可能であるが、現在までに実装したタスクではパターン照合を用いている。

図4はパターン照合によってメッセージを解釈し、情報を抽出する例を示したものである。

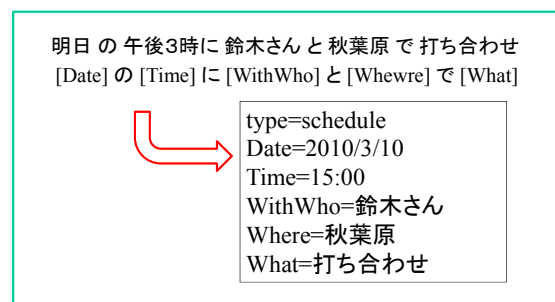


図4: パターン照合によって解釈されたメッセージ

¹ <http://xmpp.org/>

² <http://www.igniterealtime.org/projects/spark/>

³ <http://www.google.com/talk/>

⁴ <http://www.apple.com/macosx/what-is-macosx/ichat.html>

図4の例では、スケジュールタスクがメッセージからスケジュール情報を抽出している。本システムでは、このような情報を保存するためにKey=Valueスタイルのデータストア(KVS)を使用する。

KVSの利点は、リレーショナルデータベースのように予めデータ構造を決めておく必要がないことである。つまり、データ構造をタスク側で自由に決めることができる。

4.3 データストアからの情報の取得

データストアからデータを取り出すにはKey=Value形式のフィルタを使用する。図5の例では、「明日の予定は？」というユーザのメッセージを解析してtype=schedule、Date=2010/3/10という条件をフィルタにセットしている。これにより2010年3月10日のスケジュール情報をデータストアから取得することができる。

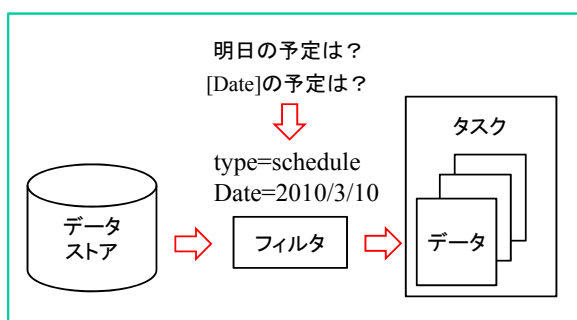


図5: データストアからの情報の取得

5. 対他ユーザ処理

5.1 セキュリティ属性の設定

本システムは、他の秘書ボットとの間で情報の交換を行う機能を備えている。情報を交換する相手に開示したくない個人情報や、業務上の機密情報を保護するために、データストアに保存された情報に対してセキュリティ属性を設定する必要がある。たとえば、あるスケジュールデータに対して「オフィシャル」のセキュリティ属性を設定した場合、他のユーザの秘書ボットからスケジュールの問い合わせを受けた際にそのボットが「オフィシャル」の権限を持っているときのみ、このデータを提供する。セキュリティ属性が設定されていない情報は、本人以外には開示されない「秘密」情報として扱う。

5.2 他のユーザの秘書ボットへの権限の付与

本システムでは、他のユーザを「Person」オブジェクトとして扱い、そのオブジェクトの属性として秘書ボットのID(アドレス)やセキュリティ権限を持たせている。

権限の種類には、仕事上の情報を共有するための「オフィシャル」や個人的な情報を共有するための「プライベート」などの他、特定のユーザグループを指定することもできる。

ユーザには、複数の権限を割り当てることができる。たとえば、「オフィシャル」と「プライベート」の権限が割り当てられたユーザに対しては、「オフィシャル」のセキュリティ属性の情報と「プライベート」のセキュリティ属性の情報の両方が開示される。

5.3 ボット間通信

ボット間通信は、ある秘書ボットが別の秘書ボットから、データストアに保存された情報を取得するための通信である。秘書ボット同士の通信もIMを利用して行うが、通信の内容は自然言語ではなく、ボット同士が簡単な手順でコミュニケーションするための定形のコマンド文を使用する。図6はボット間通信の手順を示したものである。

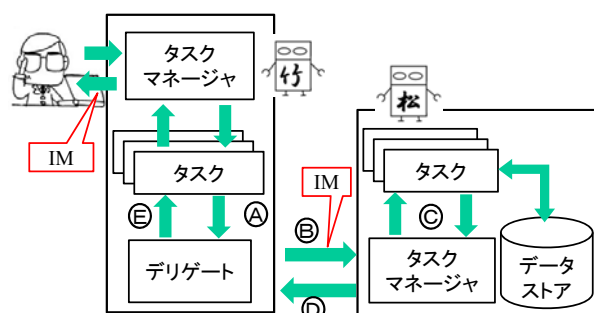


図6: ボット間通信

タスクがメインユーザのメッセージを解釈した結果、他のボットへの問い合わせの必要が生じた場合、タスクは処理を外部に委託するためのデリゲートオブジェクトを生成して問い合わせコマンドを投入する(図6中のA)。相手のボットはデリゲートオブジェクトからコマンドを受信すると、対他ユーザ用のタスクマネージャを生成する(同B)。対他ユーザ用のタスクマネージャはコマンドをタスクに転送し、応答文を受け取る(同C)。応答文はデリゲートに返され(同D)、デリゲートは自分を生成したタスクに応答文を返す(同E)。その後の処理は通常のタスクと同様である。相手ボットとの通信をデリゲートが代行するため、他ボットへ問い合わせを行ったタスクは返信を待っている間も他のメッセージを処理することができる。

5.4 相手ボットの権限に応じたデータの提供

タスクが他のボットからの問い合わせに対して応答する場合、相手ボットのセキュリティ権限と提供するデータのセキュリティ属性を照合する

必要がある。

本システムでは、タスクが對他ユーザ処理を行う場合、自ロボットが相手ロボットに予め付与している権限に応じたセキュリティの条件を自動的にフィルタに付加する。図7に示すように、タスクがフィルタを通して抽出できるデータは、相手ロボットの権限でアクセスできるものだけに限定される。

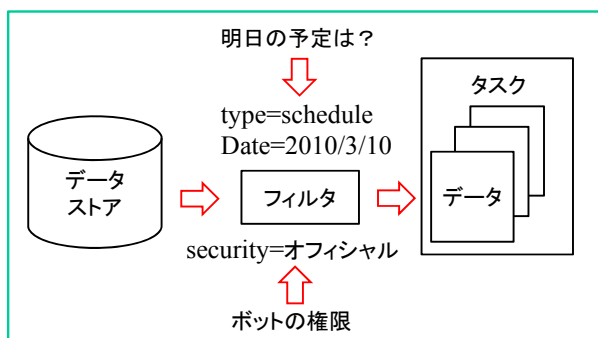


図7:相手ロボットの権限によるデータのフィルタリング

ただし、スケジュールなどの情報では、権限のないデータを単純に除外してしまうと、その時間には「予定がない」という誤った情報を相手に伝えてしまうことになる。現在はスケジュールデータに限り、日付と時間以外を空欄にした状態でフィルタを通過させ、相手には「所用」と伝えることでこの問題に対処している。

6. 実装したタスクと動作例

現在までに「挨拶」、「スケジュール管理」、「ToDoリスト」、「知人管理」、「簡易質問応答」の5つのタスクを実装し、動作テストを行っている。挨拶タスクは「おはよう」や「こんにちは」などの挨拶に応答するもので、接続の確認などに使っている。スケジュール管理は、あらかじめ登録したパターンと照合することにより、「日付」、「時間」、「場所」、「会う人」、「すること」の5項目をメッセージから抽出している。「8時から」などの時間指定に対しては、それが午前か午後かを聞き返すなどの処理も行っている。知人管理タスクでロボット間通信に必要な情報と相手のロボットの権限を設定し、他のユーザの予定をロボット間通信によって問い合わせることも可能になっている。ToDoリストタスクは「～しないと」などの文で項目を抽出し、「やることは？」などの文に回答してToDoリストを表示する。簡易質問応答タスクでは、「○○って何？」という文に対してWikiPediaの見出し語を検索し、概要の最初の段落を返す。次にユーザとロボットの対話例を示す。

```
User> おはよう
Bot> オハヨウゴザイマス。
User> 明日の予定は？
Bot> 午後5時から会議デス。
User> 金曜日は午後3時から5時まで秋葉原で鈴木さんと打ち合わせ
Bot> 3月12日ノ午後3時ニ打ち合わせノ予定ヲ登録シマシタ。
User> 金曜日の打ち合わせはオフィシャル
Bot> 3月12日の午後3時の打ち合わせヲオフィシャルニ設定シマシタ
User> 松井さんを登録して
Bot> 松井さんヲ登録シマシタ
User> 松井さんのロボットアドレスは
Bot> matsubot@example.com
User> 松井さんノ、ロボットアドレスヲ、matsubot@example.comニ設定シマシタ
User> 松井さんの明日の予定は？
Bot> 聞イテミマス
Bot> 午後5時から会議、午後7時から所用トノコトデス
User> ウミウシって何？
Bot> ウミウシとは後鰓類中で、貝殻が縮小、体内に埋没、消失などした種の総称。・・・トノコトデス。
```

7. おわりに

本稿では、チャット形式で対話を行い、人間に代わって情報の収集や管理を行う秘書ロボットシステム「YouBot」について述べた。

本システムの特徴は、ユーザの質問に対する回答をその秘書ロボット自身が持っていない場合でも、ロボット間通信を用いて他の秘書ロボットから情報を収集し、回答できる点である。秘書ロボット間で情報を交換する際にはセキュリティの確保が問題となるが、本システムでは秘書ロボットの持つ情報にセキュリティ属性、相手の秘書ロボットにセキュリティ権限を付与し、それらを自動的に照合する機能を備えている。

今後の課題は、タスクモジュールの拡充と、複数のタスクの処理結果から最適なものを選ぶための方法を検討することである。

参考文献

- [1] J. Weizenbaum. Eliza—a computer program for the study of natural language communication between man and machine. Communications of the ACM, v. 9 n. 1:36-45, 1966.
- [2] R. Wallace. Parsing the Turing Test, chapter 13. The Anatomy of A. L. I. C. E., pages 181-210. Springer Netherlands, 2008.
- [3] A. Nguyen. An agent-based approach to dialogue management in personal assistants. In Proceedings of IUI-2005, pages 137-144. ACM Press, 2005.
- [4] P. Berry, K. Myers, T. Uribe, and N. Yorke-Smith. Task Management under Change and Uncertainty: Constraint Solving Experience with the CALO Project, Proceedings of CP'05 Workshop on Constraint Solving under Change and Uncertainty, Sitges, Spain, October 2005.
- [5] A. S. Rao and M. P. George. BDI-Agents: From Theory to Practice. In Proceedings of the First International Conference on Multiagent Systems, pages 312-319, 1995.