

多重文脈に即応的な対話インターフェース：半可通

田中久美子 本村陽一 橋田浩一
電子技術総合研究所

1 はじめに

携帯電話やカーナビシステムなど、ユーザの入力手段が制限されているにもかかわらず、多くの文脈を扱う機械が増えている。たとえば、携帯電話機は年々小さくなり、ユーザは少数のボタンで全入力を余儀なくされる反面、これらの機器で扱う文脈はiモードなどの普及により増加している。これらの機器類においては、特に文脈選択の観点から、対話インターフェースは今までに重要な役割を果たす。

一方、これらの機器類で現在用いられている対話インターフェースは、文脈はユーザとの対話に基づいて決定されることが多い。その中でも大半を占めるのが、対話の手続き的な流れを予め設計者が規定し、システムがユーザを誘導して対話を行うことにより目的処理に到達する機構を用いるものである。これは、目的処理に到達するためにユーザは多段の対話を行う必要があることを意味する。さらに、ユーザはある目的処理のための対話を始めてしまうと、別の目的処理に簡単には変更できない。変更の際には、一度現在の対話をすべてキャンセルし、初期状態に戻り、改めてシステムの誘導にしたがって対話内容を変更することとなる。さらに、システム設計者は対話の手続き的な流れを設計しなければならないが、汎用な対話の手続きを設計することは、設計者にとって負担である。

これらの問題点は、対話の流れを予め規定することを廃止し、文脈を自動で選択するように対話システムを実現することで解決する。本稿では、文脈の自動選択の手法を提案し、これに基づく即応的な対話インターフェースの実装について報告する。

2 文脈の自動選択

2.1 目的とする対話

図1は現状の携帯電話機などの上の対話システムの例を示しているものとする。この図では、構造上の矢印に沿って対話が行われる。たとえば、「銀行振替」を行う場合には、

- 1(システム) 御用は何? 電話? その他の処理?
 - 2(ユーザ) その他の処理。
 - 3(システム) 番号データ? 銀行? アラーム? 給与?
 - 4(ユーザ) 銀行。
 - 5(システム) 振込? 振替?
 - 6(ユーザ) 振替。
 - 7(システム) 何銀行? ... 振替のための処理...
- と、まずはシステムに誘導されて「銀行振替」に到達するための対話を行った上で、「銀行振替」についての具体的な対話を行うこととなる。さらに、「銀行振替」の途中で緊急に電話をかける場合には、
- 8(ユーザ) キャンセル。
 - 9(システム) 御用は何? 電話? その他の処理?
 - 10(ユーザ) 電話。
 - 11(システム) 番号問合せ? 発信?
 - 12(ユーザ) 発信。

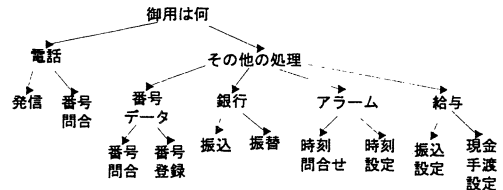


図1: 従来の対話システムの構造図

13(システム) 何番? ... 電話発信のための対話...

などと、現状をキャンセルし、改めてシステムに誘導されて「電話発信」にたどりつくための対話を行うこととなってしまふ。同じことが「電話番号登録」と「発信」でも問題となる。図では電話番号の登録と発信という関連する事項が別の構造下となっているため、登録直後に同じ番号に電話をかけるには、対話の初期状態に戻り、改めて「電話発信」を選択するための対話を行うこととなってしまふ。このように、予め対話手続きが規定されていると、ユーザは対話システム設計者の手続き設計に振り回されることになる。

しかし、図においてそもそも「銀行振替」の対話を行う対話部分は唯一しかないのであるから、何も「御用は何」の後に「その他の処理」かどうかをユーザが選択しなくとも、システムはユーザが「銀行振替」を目的としていることさえ入力されれば、即座にその対話に移動することが原理的には可能であるはずである。一方、ユーザ入力「電話番号問合せ」の場合には即座に移動することはできない。なぜなら、同じ「電話番号問合せ」が複数存在するため、どちらに移動するべきか決定できないからである。しかし、そもそもこれらの「電話番号問合せ」は本来同じものである。とすると「電話」関連の文脈としてまとめるべきであり、それを行うとシステムはやはり「電話番号問合せ」に即座に移動することが可能となる。

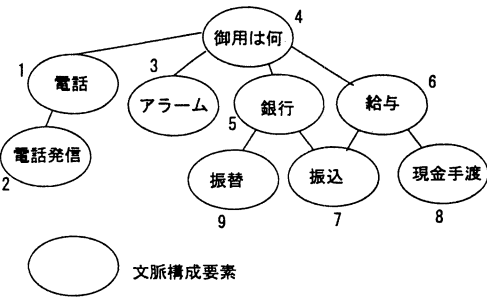
そこで、図1を対話処理の共有関係から整理して再構成することを考える。まず図の構造を同じ処理を行うかどうかでいくつかの部分に分解し、同じものはすべて共有する。たとえば、図1に複数存在する「電話」に関する部分や、「振込」は同じ処理なので一つにまとめ、共有する。つぎに、目的処理に到達するための部分を排除する。たとえば、「その他の処理」といった部分をなくす。さらに、「振替」や「振込」は「銀行」で行う対話情報を参照するから、「銀行」より下位に位置させる。これにより図1から図2が作られる。

これを用いると、同じ内容は共有されているから、ユーザのひとつで構造のどこに移動するべきかが限定され、システムは移動先を自動で決定できることとなる。たとえば、同じ例の対話は、

- 1(ユーザ) 銀行振替
- 2(システム) 何銀行? ... 振替のための処理...
- 3(ユーザ) 電話発信。
- 4(システム) 何番? ... 電話発信のための対話...

と13段の対話を4段に短縮することができる。

図2は半順序構造を成している。このノードの丸囲みを以下では、文脈構成要素というものとする。文脈とはこの半順序構造の上限ノードからつながる部分全順序構造をいうものとする。たとえば、図の4-1-2や4-6や4-6-7などが文脈である。



4-1-2, 4-5-9, 4-6, 4-6-7 文脈の例

図 2: 文脈構成要素と文脈

2.2 競り方式による文脈決定

上記の対話例では、ユーザ入力に対して答えるべき文脈構成要素が、即応的に決定されていることになる。これは競り方式を用いて実装することが可能である。たとえば、ユーザの「銀行振替」の入力に対して、まず 4 は 1, 3, 5, 6 に対してどの程度内容が合致しているかを問合せる。各文脈構成要素にはその対話で使用される文法などの情報があるので、これをもとに、ユーザ入力に対する合致度を 4 に知らせる。すると、文字列の一致具合などから、5 が選ばれることとなる。さらに、5 は 7 と 9 に同じように競りを行い、9 が選ばれることとなる。9 から 2 への移動も同様である。9 は「電話発信」とユーザに入力されても処理することができない。処理不可能な場合には、自分の前の文脈構成要素に処理が差し戻されるとしておけば、5 に対話は移る。5 も「電話発信」は処理不可能なので、4 へ移動する。4 では改めて「電話発信」を競りにかけると、1 が勝ち、さらなる競りで 2 へ移動する。ユーザは 9 においてひとこと「電話発信」といえば、2 へのこの移動は自動で行われる。

図 3 に示す構造を 1 段の競りモジュールと呼ぶ。競りモジュールの中には、半順序構造において上位の議長役と、下位の参加者がある。たとえば、前述の図 2 においては、5 は議長であり、9 や 7 は参加者である。議長は参加者を兼ねることができる。たとえば、5 は 4 が議長の時の競りにおいては参加者である。

1 段の競りは、半順序構造をより深く探索する多段の競りに拡張することができる。たとえば、4 が 1, 3, 5, 6 を選ぶ際に、5 が 7, 9 に競りを行ってからその結果を含めて 4 に合致度を知らせる場合には、「2 段の競り」となる。この場合には、「銀行振替」ではなく、「振替」だけをユーザが述べたとしても、7 に即時に到達可能となる。同様に一回の競りに対してその議長以下すべての文脈構成要素が競りに参加する方法が考えられる。

一般に、競りが何段になっても、一議長に対して複数の参加者が存在する。これを、ユーザ入力と各参加者が担当する文脈との合致度の結果を逐次に議長に通知するように実装することも可能である。しかし、参加者と議長とがまったく対称な位置付けにあることに注目すると、競り方式は並列実行に本質的に向いている。そこで本稿では、文脈エージェント（一種類の文脈構成要素を処理するスレッド）を複数用いたマルチエージェント対話を実現するものとする。この場合、図 2

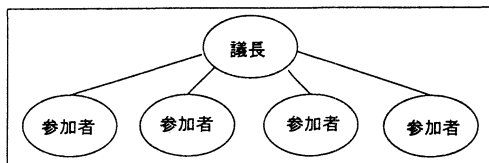


図 3: 1 段の競りモジュール

の丸囲みが、文脈構成要素を割り当てられたエージェントである。エージェントは、ある時は参加者として、またある時は議長として働く。さらに、図 2 の「振込」エージェントのように、参加者は複数の議長を持ってもよい。

3 半可通スクリプト

前節で述べた競りによる文脈決定は、対話の内容に依存せず利用できる方式である。したがって、競りを行う汎用システムと、対話に特化した文法などのスクリプトを分離して用意することができる。そして、前者は後者をシステム起動時に読み込むことにより、さまざまな対話を簡単に実現することができる。ここで、文脈構成要素間移動を決定する競りは汎用システム内で実現されるから、対話システム設計者は対話の手続き的な流れをスクリプトの中で記述する必要がない。スクリプトでは、文脈構成要素とその関係についての記述すればよいこととなる。本節ではこのスクリプトについて説明する。図 2 における 4, 5, 6, 7 の文脈構成要素のスクリプトの例を図 4 に示す。

文脈構成要素は、

1. 名前
2. 議長の文脈構成要素名の集合
3. キーワードの集合
4. 文の集合

の四つの項目から成る。文脈構成要素を担うエージェントは一つの文脈構成要素に関するスクリプトを読み込んでユーザと対話を行ったり、競りを行ったりするモジュールということになる。

文脈構成要素の名前は文脈間の関係を記述するのに用い、議長となる文脈の名前を参加者が参照することによってその関係が定まる。図 2 における「御用」「銀行」「振込」の関係は、図 4 の 8, 19 行目において示されている。この文脈構成要素間の関係は、参加者がそれまでの議長がユーザとの対話で得た情報を共有することを示している。

つぎにキーワードは、競りにおいてエージェントが自分の文脈構成要素とユーザ入力との程度合致するかを評価するために現状では用いられる¹。一方、文は対話を行なう際に用いられる。半可通ではこれらは、BNF 形式で記述するものとした。むろん、さまざまな記述形式が考えられるが、BNF 形式は一般にシステム設計者の間では広く使われている点が第一の理由である。また、半可通は音声対話システムであるが、音声認識の文法規定方法としてしばしば BNF が用いられている [2][4] こともその選択理由の一つである。

とはいえ、対話を進める関係上、BNF をつぎの点で工夫した。まず、第一に、目的処理を行うためにはシステムがユーザから得なければならない情報がある。これをスロットと呼ぶ。そして、スロットは疑問用と答用の二つの BNF 形式の組として定義することで、登録と問合せの文章をなるべく共有して規定できるようにした。たとえば、図 4 の 6 行目は、「誰」という項目を埋めるスロットであり、21, 22 行目を用いると、「(田中の) 給与は 100 円です」といった文章が生成あるいは認識される。

第二に議長文脈と共有されるスロットは、括弧内に書くものとした。たとえば、

```
Context Parent {
  Phrase=SlotPerson と SlotPerson です;
  ###          1          2
}
Context Child {
  Upper = Parent;
  Phrase=(SlotPerson) の SlotPerson です;
  ###          3          4
}
```

¹評価関数の定義の仕方によっては、必要ない。

```

###4 の定義 (行頭の#はコメントを表す)
1 Context Goyo {
2   Keyword = goyo;
3   Phrase = SlotPerson, goyo は 何;
4 };
5 goyo = 御用;
6 SlotPerson = (誰, 田中|本村|橋田);
###5 の定義
7 Context Bank {
8   Upper = Goyo;
9   Keyword = bank|account;
10  Phrase = (SlotPerson の) SlotBank の account
11         は SlotAccountNumber です;
12 };
13 SlotBank=(何銀行|どこの銀行, bankName bank);
14 bank = 銀行;
15 bankName = 富士 | さくら | 第一勧業;
16 account = 口座番号;
17 SlotAccountNumber = (口座は何番, [0-9]*);
###6 の定義
18 Context Salary {
19  Upper = Goyo;
20  Keyword = salary;
21  Phrase = (SlotPerson の) salary
22         は SlotAmount です;
23  Phrase = (SlotPerson の) つぎの
24         salary は SlotWhen に 出ます;
25 };
26 SlotMoneyAmount == (いくら, [1-9]+[0-9]* 円);
27 SlotWhen = (いつ, date);
###7 の定義
28 Context BankPayment {
29  Upper = Salary | Bank;
30  Keyword = 振込;
31  Phrase = SlotBank の SlotAccountNumber に
32         (SlotAmount を) 振込ます;
33 };

```

図 4: 半可通スクリプトの例

の場合には、3 は 1 か 2 を共有する。どちらであるかは、システムが 1 か 2 のいずれかで 3 を埋めてユーザに確認する。一方、4 は 1 も 2 も共有しない。スロットの共有は、同じ文脈内でしか起こらず、上位の文脈構成要素のスロットを下位の文脈構成要素が参照するにとどまる。スロットが省略可能の場合も、共有の場合も、システムにとって曖昧性が含まれてしまうため、その内容の確定にはユーザに確認の上対話を進めていくものとした。

4 半可通システム

本システムのエージェントは、上記スクリプトを読み込み、ユーザと対話、競りによる文脈間の移動、の二つの処理を行う。以下、順に説明する。

4.1 エージェントによる対話

競りに勝って、対話権限が与えられると、エージェントはスロットを用いて対話を行なう。スロットは、システム内でその優先順位が計算され、その順位にしたがって、埋められる。ユーザの入力が、優先順位を変更するようなものである場合には、それに従う。

ユーザがスロットの現行値を変更したい場合には登録を明示的に指示する。問合せやシステムに対する実行命令²の場合には、システムはデータベース³に問合せる。登録の場合には、スロットを埋める対話を続け、データベースに登録する。さらに、ユーザ入力が自分の持つ文法に合致しない時には、対話は破綻する。対話が破綻するか終わると、次節でのべる競りを行う。

4.2 文脈構成要素間の移動

ユーザとの対話権限を持つエージェントが、競りを主催することができる。このエージェントは、以下の処理を順に行う。

1. 対話権限を議長から受けとると、ユーザとの対話を行う。

²銀行振り込みなど実際の実行などを表す

³現状では単なる連想データベースである。

表 1: 対話例

step	対話文	対話権者
1	誰?	4
2	田中	4
3	田中、御用は何。	4
4	給料問合わせ	4 → 6
5	田中の給料は 100 円です。	6
6	いつ?	6
7	田中のつぎの給料は 2000 年 1 月 1 日に 出ます。	6
8	振込。	6 → 7
9	何銀行?	7
10	さくら銀行	7
11	口座は何番?	7
12	1 2 3 4 5 6 7	7
13	さくら銀行の 1 2 3 4 5 6 7 に 100 円の 振込を行いますか?	7
14	はい。	7
15	さくら銀行の 1 2 3 4 5 6 7 に 100 円の 振込を行います。	7
16	銀行	7 → 6 → 4 → 5
17	田中の富士銀行の口座番号は 3 4 5 6 7 8 です。	5
18	振込。	5 → 7
19	何銀行?	7
20	第一勧業銀行	7
21	口座は何番?	7
22	9 8 7 6 5 4	7
23	第一勧業銀行の 9 8 7 6 5 4 に 100 円を 振込ますか?	7
24	いいえ。	7
25	第一勧業銀行ですか。	7
26	はい。	7
27	9 8 7 6 5 4 ですか。	7
28	はい。	7
29	1 0 0 円ですか。	7
30	1 億円。	7
31	第一勧業銀行の 9 8 7 6 5 4 に 1 億円を振 込ますか?	7
32	はい。	7
33	第一勧業銀行の 9 8 7 6 5 4 に 1 億円の振 込を行います。	7
34	電話	7 → 5 → 4 → 1
35	誰?	1
36	本村	1
37	1 0 2 4 です。	1 → 2
38	かける。	2
39	1 0 2 4 に電話をかけますか。	2
40	はい。	2
41	1 0 2 4 に電話をかけます。	2

2. 対話が終了あるいは破綻すると、まずは自分が議長となつて自分で競りを開く。競りにおける参加者は、スクリプトの中で自分を Upper として参照するすべてのエージェントである。
3. 競りの結果、勝った参加エージェントがいる場合には、対話権限を渡す。
4. 勝ったエージェントがいない場合には、自分の前に議長であったエージェントに対話権限を差し戻す。

対話権限が上位文脈から下位文脈へと移動するのは、競りによる場合に限られる。対話権限が下位文脈から上位文脈に移動するのは、対話と目的処理が終了するかユーザとの対話が破綻するかし、さらに自分を議長とする競りで勝った参加者がいない場合である。

競りを行なうには、ユーザの入力に対する合致度を計算する評価関数が必要である。この評価関数の設計は、さまざまに考えられる。たとえば、キーワードを元に一致文字数とすることも可能であり、現状の半可通はこの簡単な方式を採用している。また、半可通では、音声認識を各エージェントの機能の 1 つとしてもたせており、音声認識器が出力する認識結果の確からしさを評価値として用いることも可能である。競り方式を用いる限り評価関数は動的に変更可能であるため。

ユーザごとの履歴に基づく評価関数を用いる方式を採用することもできる。

5 実験

図4に示すスクリプトを読み込んだ半可通との対話例を表1に示す。この例では、給与の間合せとその振込み指定を行ない、さらに1億円の銀行振込を指示し、その後、本村に電話をかけるという対話例となっている。対話権限者であるエージェントの番号は図2中の番号に対応する。尚、システムの出力はこのフォントで示す。

さらに、半可通システムの効率をつぎのように示す。ある携帯電話上で扱うことのできる全文脈(図1に対応するものとする)と、それを半可通で実現した時(図2)の対話段数を仮想的に比較する。対話の1段とは、ユーザあるいはシステムの一回の入力をいうものとする。図5は、文脈の数を横軸とし、その文脈をこなすのに必要となった対話段数を縦軸に描いたものである。文脈間移動のステップ数が減る分、対話の効率が半可通の場合の方が高く、少ない段数で対話を終わることができることがわかる。

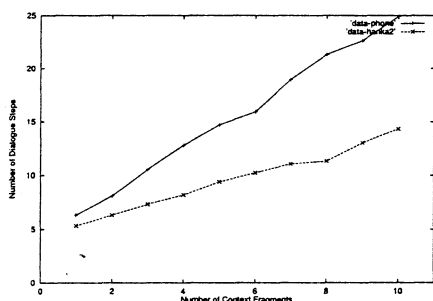


図5: 半可通システムの効率

6 関連研究

まず、本稿の問題意識を共有する論文は他にもすでに存在する。本研究は、松井らによる事情通ロボット[5]の対話システムを起点としているが、同研究者の間でも本稿と同様の問題意識に基づく指摘はある[8]。

つぎに、スロットに類似したデータ構造を用いた対話システムについてはこれまでも提案がある。たとえば、[6]はユーザが埋めなければならない項目をformと呼び、formにユーザ入力との合致度を計算する機構を持たせている。合致度の高いformから順に埋められるようになっている。formの合致度の計算方式は、スロットを埋める順序を決定する際に参考となるものである。

ユーザが主導権をもって対話を行う研究は、システム主導のそれに比べるとまだ少ないといえる。その一つの原因はユーザ主導で複雑な対話を行うには、ユーザの意図やプランをシステムが推測する必要があり、そのための基礎研究に焦点が当てられるためと考えられる。そもそも、これまでの対話研究は、分野を限定し、その上でいかに複雑な対話を実現するかという方向で研究が進められて焦点が当てられており[3]、本稿のように、個々の対話は簡単にとどめる代わりに複数の文脈を扱おうという試みはこれまでにはあまりみられないとの印象を受ける。

対話における主導権については、Suzukiら[7]は、目的処理からgoal-orientedに文脈を決定するのではなく、人間とシステムが対話をするうちに徐々に文脈が決定されていくとするマルチエージェント対話システムを提案した。したがって、文脈はユーザとシステムの両者が交互に主導となって進められる点で本手法には類似している。

マルチエージェント対話システムの観点では上記の研究の他にも数多く存在する。中でも[1]らは、異なる話題を割り当てられたエージェント間の内容が矛盾したり衝突したりした場合に、ユーザに対する答えを各エージェントのプランに基づいて協調して呈示する方法について示している。この点は、本稿における評価関数の代替案として参考となる。

7 結論

本稿では、複数の目的処理を扱うにも拘らず、入力手段が制限されているような機械における対話システムの実現法について論じた。従来の対話システムでは対話の手続き的な流れが規定されており、この方式下では目的処理に到達するために対話を行う必要がある。すると、目的処理を変更するには、現状の対話をキャンセルし、改めて目的処理に到達するための多段の対話を行うことになってしまう。

本稿で提案する半可通では、ユーザ入力に対してシステムが自動で文脈を決定する。そのためには、対話手続きの上で、同じ対話内容が複数分散して存在している、システムは文脈を一意に決定することができない。そこで、現状の対話の手続き的な流れを文脈の構成要素に分解し、同じ文脈が唯一となるように再構成した。その上で、文脈の構成要素を担うエージェントがユーザ入力に対して競りを行ってつぎの対話権限を争い、勝ったエージェントがユーザとの対話権限を得ることによりシステムは自動で文脈を決定する。この方式により、文脈間の移動がユーザのひとことで行うことができ、即応的な対話が可能である。

半可通は上記のように再構成された文脈構成要素の規定により初期化される。すなわち、システム設計者は対話の手続き的な流れを規定する必要はなく、文脈に関する簡単なスクリプトを記述するだけでよい。本稿ではこの半可通スクリプトの記述方法と、半可通システムの機構について示した。最後に、本対話システムとの対話例と対話効率に関する実験結果を示した。

今後の課題であるが、第一に競り方式に関して多段の競りやユーザモデルの導入などの課題があるため、これに関してシステムを変更し実験を行うことが挙げられる。特にユーザモデルに関しては、画像処理や音声認識では画像処理や音声によるユーザ認証が昨今では可能となってきたため、本システムとの融合させて用いることが考えられる。第二に、受け付けシステムや音声携帯電話、さらにロボットへの応用などさまざまな応用を試みて、使用感の実験を行うことが挙げられる。

参考文献

- [1] J. Chu-Carroll and Sandra. Carberry. Collaborative response generation in planning dialogues. *Computational Linguistics*, 1, 1999.
- [2] 橋戸 et al. Rex 使用説明書. NTTのソフトウェア, 1, 1998.
- [3] J. Glass. Challenges for spoken dialogue systems. In *Proc. 1999 IEEE ASRU Workshop*, 1999.
- [4] K. Itou, S. Hayamizu, K. Tanaka, and H. Tanaka. System design, data collection and evaluation of a speech dialogue system. *The Institute of Electronics, Information and Communication Engineers of Japan, Trans. Inf. & Syst.*, E76-D(1):pp.121-127., 1998.
- [5] T. Matsui and et al. Asoh. Integrated natural spoken dialogue system of jijo-2: Mobile robot for office services. In *Proceedings of AAAI-99*, 1999.
- [6] K.A. Papineni, S. Roukos, and R.T. Ward. Free-flow dialog management using forms. In *Proceedings of Eurospeech '99*, 1999.
- [7] N. Suzuki, S. Inoguchi, K. Ishii, and M. Okada. Chatting with interactive agent. In *Eurospeech '97*, volume 3, pages 2243-2247. 1997.
- [8] 高木朝, 中島秀之ほか. JDT: 日本語対話システム構築用ツール群の開発プロジェクト. In 人工知能学会言語・音声理解と対話処理研究会, 1999.