ヒューマンインタフェース 80－1
音声言語情報処理 23－1
（1998. 10. 16）

スクリプト言語を用いたマルチモーダル対話記述の試み

秋葉友良 伊藤克亘

電子技術総合研究所

システム設計者が、人と機械の間のマルチモーダル対話を容易に設計しテストすることができる枠組みについて述べる。MILES(Multi-modal Interaction LEading Script)は、マルチモーダル対話システムのための対話モデルを記述する言語である。MILESは、ユーザにとって自然で役に立つマルチモーダル対話を実現するために、対話中のイベントの時間関係と割り込みなどを始めとする対話性について注意を払って設計した。MILESで記述したスクリプトは、アプリケーションプログラムとは切り放して実装された対話管理モジュールによって解釈・実行される。対話管理の領域依存性はアプリケーションプログラム毎に用意されたスクリプトによって表される。したがって、提案する枠組みはどんな領域のマルチモーダル対話も、システムの再構築なしに、動作させることができる。本稿では、MILESで記述を試みたいくつかのマルチモーダル対話タスクについて報告する。

An Attempt to Design Multi-Modal Interactions using a Scripting Language

AKIBA Tomoyosi, ITOU Katunobu

Electrotechnical Laboratory

We introduce the architecture that gives system designers the easy way to design and test the multi-modal interaction between a human and an artifact. We propose a programming language MILES (Multi-modal Interaction LEading Script) with which system designers can easily design the dialogue model for multi-modal interaction system. The language was carefully designed so that it can deal with both the precise time relation between events of any mode and the communicative elements, like interruptions in spoken dialogues, in order to make the multi-modal interaction go smoothly. We call the dialogue model written in MILES a 'script'. A script is directly processed by the interpreter that works as the dialogue management component. The system is implemented apart from the application program. Because the domain dependent aspect of the dialogue model can be written through the script, our system is able to work for any domain without reconstruction. In this paper, we will report an attempt to design several multi-modal interactions using MILES.

1 はじめに

筆者らは、複数のモードを通した人と計算機との間の対話について研究を行っている。これまで、いくつかの領域(道案内[1]、電子秘書[2]、論争支援[3])についての対話システムを構築してきた。これらのシステムの対話に関する処理を担当するモジュールは、領域(アプリーケーション・プログラム)に大きく依存しおり、個別に設計・実装されている。

しかし、新しい領域の対話システムを構築する度に対話モジュールを一から設計し直すことは、システム開発のコストを考えた場合大きな負担となる。さらに、複数のモードを用いた複雑な対話処理が要求されるマルチモーダル対話システムにおいて、その対話処理のプログラミング自体が大きな負担となる。

そこで我々は、対話に関する処理だけを独立に記述する言語を設計し、それを解釈実行するモジュールをアプリケーション・プログラムとは切り放して実装することを試みた。この枠組みによって、アプ

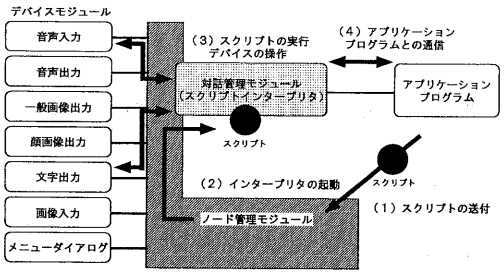

図 1 システムの構成と実行の手順

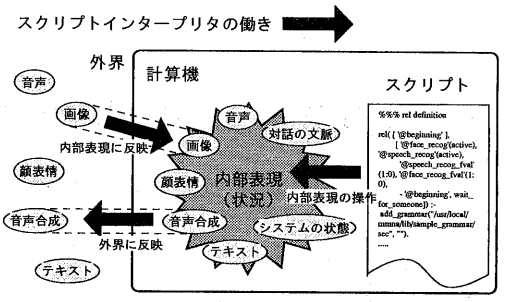

図 2 インタープリタ

リケーション・プログラムに大きく手を加えることなく対話に関する処理だけを独立に開発することができる上、対話に関する資源の共有や再利用が容易となり、システム開発のコストが軽減されるはずである。

MILES[5]は、このような目的で設計した言語である。本稿では、MILESによって記述を試みた5つのタスクについて報告する。

2 マルチモーダル対話記述言語 MILES

我々のマルチモーダル対話システムは、目(見る)、耳(聞く)、口(話す)、そしてCGエージェントによる顔表情(笑う、悲しむ、怒るなど)といったモードを合わせ持つ。これらの機能は、それぞれ、画像認識モジュール、音声認識モジュール、音声合成モジュール、顔画像合成モジュール、によって実装されている。システムは、これらのデバイス・モジュールに加え、モジュール間の通信を一元的に管理するノード管理モジュール、ユーザとの対話を担当する対話管理モジュールから構成される(図1)。

これらのモジュールは、領域依存の処理を担当するアプリケーション・プログラムとは独立して実装されている。ユーザとの対話によって情報を獲得しようとする(任意の)アプリケーション・プログラムは、その手順を記した「スクリプト」を用意し、それをシステムへ送付する。対話管理モジュールは、「スクリプト」の記述に従ってデバイス・モジュールを使用しユーザとの対話を遂行する。また、アプリケーション・プログラムは、対話管理モジュールと直接通信することができ、対話によって獲得した情報を受け取ったり、対話管理の実行の制御を行うことができる。

MILES(Multimodal Interaction LEading Script)は、「スクリプト」の記述のための言語である。システム設計者は、MILESによって、多種のモードを媒介としたシステムとユーザとの対話を記述する。MILESの設計にあたっては、特に次の2点に注意を払った。

時間関係 マルチモーダル対話では、複数モード間の時間関係を扱う必要がある。特に、入力モードの同時性、出力モードの時系列上の順序。

対話性 自由な対話の実現[4]。システム発話への割り込みを可能にする。

MILESでは、「状況」と呼ばれる内部表現の操作によって、ユーザとの対話を記述する。「状況」は、システムの内部状態、システム外部の状態、対話の文脈など、対話に関係する資源を一つの表現にまとめたものである。システムからの外部への出力、外部からの入力等は、この「状況」を介して実行される。(図2)。ユーザとの対話は、ある「状況」においてどのように「状況」を変化させるかを記述する(「状況」間の関係を記述する)ことによって、組み立てられる。

当システムは、SGIワークステーション(Indy または O2)の上に実装されている。全てのモジュールは一台のワークステーションで動作可能である(図3)。

2.1 対話記述例

現在MILESはPrologの拡張として実装されているため、記法はPrologのそれに準ずる。図4の定義は、「1秒の間に画像`Some`と音声`hello`が同時に現れ、かつ、まだ`Some`に対して挨拶を行っていない状況」を認識し、音声合成による声とCGエージェントにより挨拶を表出する。挨拶の表出が完了すると、そのことを対話の文脈に付け加える。述語`rel`の第1引数で該当する状況を指定し、第2引数で次の状況への差分を与える。

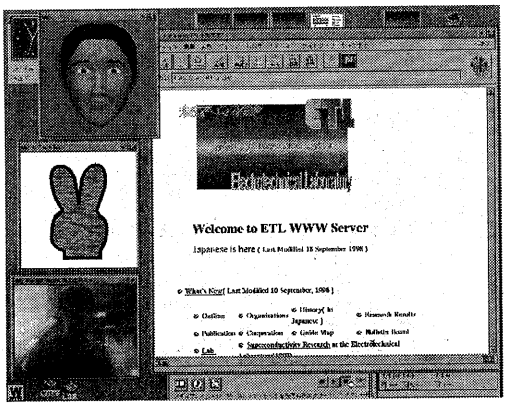

図 3 実行イメージ

```
rel( [ -greeted(Some),
       '@timeval'(1:0),
       '@face_recog_interp'(Some),
       '@speech_recog_interp'(hello) ],
     [ '@act'( [ speak("こんにちわ"),
                 +[greeted(Some)] ] ) ]
   ) :- someone(Some).
```

図 4 記述例

対話は、このような定義の束を与えることで記述される。対話全体の定義の例は、文献[5]を参照されたい。

3 タスク

前節で述べた対話スクリプト MILES を用いた対話記述を試み、5種類のタスクを記述した。

3.1 画像登録、画像認識のテスト

「画像登録」スクリプトは、顔認識に必要な画像登録を自動化するためのスクリプトである。登録を行う者をカメラの前に導き、登録開始の合図とともに首を動かして様々な角度の顔をカメラに見せることを指示する。「画像認識テスト」スクリプトは、登録した画像を使って正しく認識ができるか確認するためのスクリプトである。これらは、これまでデモンストレータが行ってきたインストラクションを、計算機上のエージェントとの対話によって自動化する。

3.2 じゃんけん

時間関係の制御を行うスクリプトの例題として、計算機上のエージェントと音声認識を使ってじゃんけんを行うスクリプトを記述した。

エージェントの掛け声(「じゃーんけーん」)とともに、ユーザはタイミング良く自分の手を発声しなければならない。画像で表示するエージェントの手よりも後にユーザが発話した場合、「後出し」と見なされ、もう一度やり直しとなる。また、発声が速すぎたり、いつまでたっても発声がない場合も、エージェントから注意が促され、やり直しとなる。タイミングよく発声が行われると、それぞれの手を比較して勝敗が決まる。

図5に示すスクリプトの断片は、このタイミングの判定を記述したものである。対話例を以下に示す。

(計算機の前に、ユーザが現れる。)
S: こんにちは、〜さん。じゃんけんをしましょう。
S: (真剣な表情) じゃーんけーん
U: ちょき
(自分の手の画像表示)
S: (驚いた表情。悲しい表情) あなたの、勝ちです。
もう一度、じゃんけんをしましょう。
S: (真剣な表情) じゃーんけーん
(自分の手の画像表示)
U: ぐー
S: (驚いた表情。怒った表情) あとだしは、ずるいですよ。

3.3 電子秘書

以前、マルチモーダル対話のデモンストレーションとして開発したシステム[2]を、スクリプトを使って再記述を試みた。カメラの前の人物を認識し、話しかけ、ユーザの質問に応じて、メールの到着を知らせ、それを読み上げたり、時刻・日付を応えたりなど、簡単な対話を行う。さらに拡張を行い、WWW上の特定のページの表示や読み上げ(ニュース、天気予報)、システム自分自身の説明にも対応した。対話例を以下に示す。

(計算機の前に、ユーザが現れる。)
S: こんにちわ、〜さん。
U: 今、何時?
S: 〜時〜分です。
U: メールは、来てる?

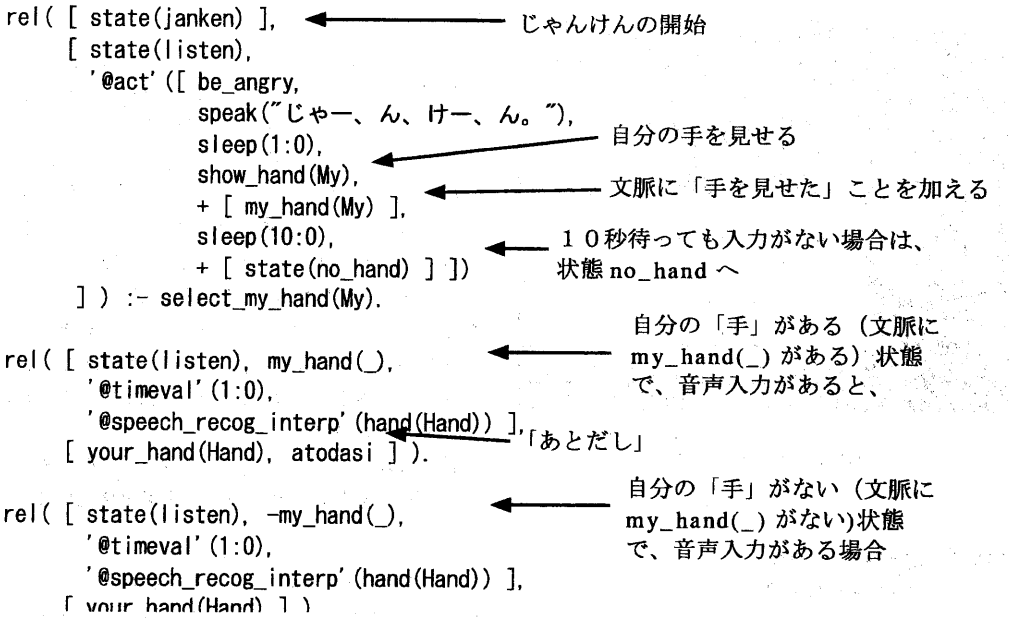

図 5 ユーザ発話のタイミングの判定

S: はい、～さんから来ています。読みますか?
U: はい。
S: 「今日の会議は中止です。」とのことです。
U: ニュースを見せて。
S: ニュースのページを開きます。
(インターネット上の、ニュースページを表示する。)
U: ありがとう。
S: どういたしまして。

3.4 地下鉄乗り換え案内

現在位置と行き先を尋ねた上で地下鉄による経路を探索する。また、提示した経路についての乗り換え案内、時間・料金の質問応答を行う。探索の範囲は、東京の営団地下鉄とした。記述した対話には、以下のような戦略を用いた。

1. 対話の状態毎に単語辞書を切り替える。
2. 割り込みや複数のモードを活用し、誤認識に対するユーザの負担を軽減する。

1の戦略によって、語彙の絞り込みにより認識率を向上させることができる。

2の戦略は、対話システムとしてのユーザビリティを向上させることを目指している。このタスクでは数多い駅名などを認識するため、音声認識時の誤認識は避けられない。一方、ユーザとの対話によって、誤認識による対話の破綻を回避することができる。例えば、認識した単語の確認の発話を行い、ユーザの応答(はい、いいえ)を求めるといった戦略が考えられる。しかし、確認の多用は、対話のテンポを損ない、利用者の苛立ちを招く恐れがでてくる。

このスクリプトで用いた戦略では、認識した単語の確認は行うが、ユーザの応答を強いることはしない。確認の発話の後、次の発話へと進む。ユーザの発話は、割り込みとして処理される(図6)。誤認識を示唆する発話(「ちがうよ」など)を認識した場合は、処理を中断して対話の状態を戻し、もう一度目的の単語の発話を促す。肯定の発話(「はい」など)を認識した場合、(音声出力とは別のモードである)顔表情によって、正しく認識したことをユーザに伝える。

以下に、対話例を示す。

(計算機の前に、ユーザが現れる。)
S: こんにちわ、～さん。今日はどこから来ましたか?
U: つくばからです。
S: 営団地下鉄の経路について御案内いたします。どこから、どこまについて、質問したいですか。

U: 上野から早稲田まではどういけばいいの?
S: 根津から早稲田までですね。いましらべますので…
U: ちがうよ。
S: まことにすみません。もう一度お尋ねします。どの経路ですか。
S: 上野から早稲田までです。
S: 上野から早稲田までですね。いま…
U: はい。
S: (笑顔で応える)しらべますので、お待ち下さい。
(経路探索)
S: 銀座線、東西線に乗ります。この経路について、乗り換え、所要時間、料金など御案内できます。何か、御質問は、ございますか。
U: 東西線は、どこで乗り換えますか。
S: 東西線の乗り換え、ですね。今調べますので、お待ち下さい。
S: 東西線は、日本橋です。他に、何かございますか。
U: どのくらいかかりますか。
S: 時間ですか、それとも値段をお尋ねですか?
U: 値段です。
S: 上野から早稲田までの値段ですね。今調べますので、お待ち下さい。
S: １６０円です。他に何かございますか。
U: ありがとう。
S: どうもありがとうございました。さようなら。

3.5 WWWブラウジング

音声認識技術を用いた、WWWブラウジングやホームページ検索の試みは、すでにいくつか報告がある[6]。ここでは、音声認識によってリンクをたどる機能のみを取り上げ、対話スクリプトでの記述を試みた。このスクリプトで用いた戦略を以下に示す。

- リンクをたどる場合は、そこに含まれる名詞あるいは名詞の連続を発声してもらう。
- どのページでも利用できる共通の発話を想定する。(「戻って下さい」など。)
- ユーザからの発話を待ちの状態で、一定時間発話が認識されなかった場合、システムのインストラクションを発話する。
- ページをたどる時は必ず確認の発話を行う(「～ですね」)。質問ではないので、ユーザの返答は待たない。
- 確認の発話の後、ユーザから認識誤りを示唆する発話(「違います」など)があった場合は、ページの移動を行わない。また、ユーザから直接リンクの名前が発話された場合、そのリンクへ修正する。

タスク	規則数	行	その他
画像登録	4	74	
電子秘書	37	627	
じゃんけん	19	238	
地下鉄乗り換え	47	665	database
WWWブラウズ	16	246	perl script

表 1 各スクリプトの記述量

リンクに含まれる単語の認識には、そのページの情報から動的に認識用辞書を構築する必要がある。これには、形態素解析システム「茶筅」[7]とperlプログラムを利用した。対話例を、以下に示す。

(ホームページの表示)
S: 情報を獲得しています。ちょっと、まってね。
(考え込む表情)
S: (笑顔で)おまちどうさま。
(しばらく待つ)
S: 移動する場合は、リンクに含まれる単語を、言ってください。
(しばらく待つ)
S: 一つ前のページに戻るときは、「戻る」と…
U: 豪雨
S: 「王」ですね。この…
U: ちがうよ
S: (驚いた顔。つづいて、悲しい顔)ごめんなさい。
U: 集中豪雨
S: 「集中豪雨」ですね。このリンクを…
U: 横浜
S: (驚いた顔)では、「横浜」ですね。このリンクをたどります。
(ホームページの表示)

4 おわりに

スクリプト言語MILESを用いて記述した、いくつかのマルチモーダル対話タスクについて述べた。この言語で扱えるモード間の時間関係の記述やシステムの表出への割り込み機能を用いて、利用者にとって親しみやすい対話の記述を試みた。各タスクの記述に要した記述量は、表1のとおりである。

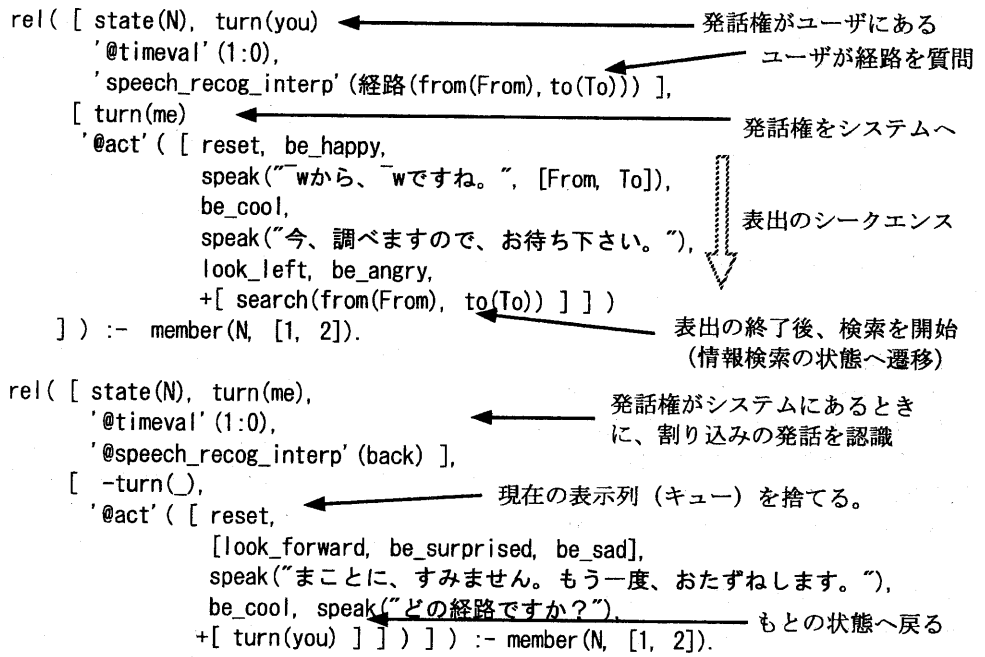

図 6 割り込みの扱い

スクリプト言語の採用により、新しいタスクの対話システム構築が容易になった。一方、記述力の限界も明らかになりつつある。例えば、現在のスクリプトでのモード間の時間関係の記述では、各モードでのイベントを時点として捉え、その同時性や前後関係を指定する。より厳密な時間関係の記述では、イベントを時間間隔として捉える必要があるだろう。言語の拡張を検討していきたい。

今回実装したタスクでは、ユーザが文相当の単位を発話することを想定している。ユーザに負担をかけない音声対話システムとしては、「文」の発話を想定しない枠組みが重要であると考えている [4]。本稿で述べたシステムに、休止間音声の言語モデル [8] を導入すれば、システム自体に手を加えることなく、言語モデルの置き換えと、スクリプト上での解釈系の修正で実現できると考えられる。今後検討していきたい。

参考文献

[1] Itou, K., et al. Collecting and analyzing nonverbal elements for maintenance of dialog. In ICSLP, pp.907-910, 1994.

[2] Hayamizu, S. et al, Multimodal interaction system at the Electrotechnical Laboratory, Proceedings of Real World Computiong Symposium, pp.16-22, 1997.

[3] 新田克己, 他. 論争支援マルチモーダル実験システム MrBengo. 電子情報通信学会論文誌 D-II, Vol.80-D-II No.8, 1997.

[4] 伊藤克亘, 秋葉友良, 上條俊一, 田中和世. 休止を区切りとした対話処理. 音声言語情報処理, 95-SLP-7-22, 1995.

[5] 秋葉友良, 神嶌敏弘, 伊藤克亘. 時間関係と対話性を考慮したマルチモーダル対話記述用スクリプト. 信学技報, NLC97-53, SP97-86, 1997-12.

[6] 近藤和弘, チャールズ ヘンプヒル. 音声認識を用いた WWW ブラウザとその評価. 信学論, Vol.J81-D-II, No.2, pp.257-267, 1998.

[7] 松本, 他. 日本語形態素解析システム「茶筅」version1.5 使用説明書, 奈良先端大学松本研究室.

[8] Akiba, T. and Itou, K. Modeling Inter-pausal Phrases towards Spontaneous Speech Dialogue Systems. Proceedings of Natural Language Processing Pacific Rim Symposium, 437-442, 1997.