LISP による知的応答システムの試作

——新幹線列車案内システム——

村野　勝巳*・金岡　泰保**・岡田　敏彦**・富田　真吾**

Shinkansen Travel Guide System by LISP

Katsumi MURANO, Taiho KANAOKA, Toshihiko OKADA
and Shingo TOMITA

Abstract

This paper describes the system "Shinkansen Travel Guide System" designed by using the symbol manupulating language LISP. This system is intended to select a desired schedule of Tokaido and Sanyo Shinkansen through the conversation between user and computer. In this system, the contents of the conversation is retricted to Shinkansen travel guide, and so the answer from user is processed by the simple language understanding part. Also, this system facilitates to correct the error inputted station names and to modify the data in the timetable.

1. まえがき

近年，従来の数値計算向き言語である FORTRAN や ALGOL 等とは趣きを異にした記号処理用言語である LISP や PROLOG が注目されており，SCHOLAR[1] や GUS[2] 等種々の情報処理システムの作成に応用されている．このようなシステムにおいては，自然で柔軟な対話形式を実現するために高度な自然言語処理機構を取り入れている．しかしながら取り扱う対象の問題領域を限定した場合は，その対話形式も制限され簡易な言語処理機構で十分目的を達成することができる．

本論文は，対話形式を制限した，コンピュータとの簡易な相互対話型システムの実現を目的とし LISP を用いて，東海道・山陽新幹線列車案内システムを試作したのでその報告を行なうものである．このシステムは利用者が新幹線の列車の選択をシステムとの対話によって行なうことができるようにしたものである．対話内容を列車案内に限定することで対話の形も大幅に制限することが可能となり，従って対話文の処理も文の中からキーワードを見つける（キーワード法）という簡易な手法を用いている．又，対話文の処理の中に駅名の誤入力の場合の処理を加え，これにより誤入力及びローマ字表記の個人差の問題をかなり解消することができた．さらに列車ダイヤの改正に伴うデータ変更を容易に行なえるようにした．

なお本システムは，本学計算機センターの ACOS-6. TSS 端末を介して，LISP を起動することにより使用することができる．

2. 応答例

Fig. 1 に本システムの応答例を示す．Fig. 1 で下線部は利用者の応答文，その他はシステムの出力である．

3. システムの構成

旅行会社が客から旅行に関する案内を依頼された場合，(1)まず，旅行に関する情報を客から聞く．(2)その情報をもとに旅行計画をたて，客にその計画の良否を問う．(3)客がその計画を気に入らなければ，旅行計画をたて直す．といったような過程を考えることができ

* 電子工学科（現，東京芝浦電気）
** 電子工学科

```
(1)   HELLO !!
      I CAN HELP YOU PLAN
       A TRIP BY SHINKANSEN.

(2)  WHERE DO YOU WANT TO GO ?
(3) =(I WANT TO GO TO TOKYOH)

(4) PLEASE SELECT A RIGHT STATION FROM THE FOLLOWING STATION LIST
    (TOKYO TOYOHASHI)
(5) =TOKYYO

(6) I JUDGE THAT THE WORD <TOKYYO> IS THE STATION <TOKYO>

(7) WHAT TIME DO YOU
        WANT TO LEAVE ?
(8) =(I WANT TO ARRIVE BEFORE 7 PM)

(9)    *** PLEASE WAIT A MOMENT. I AM THINKING .....

(10) WOULD YOU LIKE THE TRAIN
        THAT LEAVES AT 11:50 AM AND ARRIVES AT  6:56 PM ?
(11) =OK

(12) HIKARI 72 THAT LEAVES OGORI AT 11:50 AM
                      AND ARRIVES HIROSHIMA AT 12:48 AM
     PLEASE CHANGE AT HIROSHIMA
     HIKARI 142 THAT LEAVES HIROSHIMA AT  1:18 PM
                      AND ARRIVES TOKYO AT  6:56 PM

(13) SHALL I HELP YOU PLAN
         A RETURN TRIP ?
(14) =YES

(15) WHAT TIME DO YOU
        WANT TO LEAVE ?
(16) =(4 PM)

(17)    *** PLEASE WAIT A MOMENT. I AM THINKING .....

(18) WOULD YOU LIKE THE TRAIN
        THAT LEAVES AT  4:00 PM AND ARRIVES AT 10:18 PM ?
(19) =(WHAT IS THE NEXT TRAIN)

(20)    *** PLEASE WAIT A MOMENT. I AM THINKING .....

(21) WOULD YOU LIKE THE TRAIN
        THAT LEAVES AT  5:00 PM AND ARRIVES AT 10:53 PM ?
(22) =(FIRST ONE)

(23) HIKARI 81 THAT LEAVES TOKYO AT  4:00 PM
                      AND ARRIVES OGORI AT 10:18 PM

(24) ****** SCHEDULE ************************************
     *                                                  *
     *       OGORI          HIROSHIMA          TOKYO    *
     *            HIKARI                                *
     *              72 GO                               *
     *      11:50 --------> 12:48                       *
     *                              HIKARI              *
     *                                142 GO            *
     *                      13:18 --------> 18:56       *
     *                                                  *
     *       TOKYO           OGORI                      *
     *            HIKARI                                *
     *              81 GO                               *
     *      16:00 --------> 22:18                       *
     *                                                  *
     *--------------------------------------------------*
     * COST                                             *
     *   SINGLE TRIP ... ¥17000(¥ 7400)                 *
     *   ROUND TRIP ... ¥32080(¥14800)                  *
     ****************************************************

(25) SHALL I HELP YOU PLAN
          ANOTHER TRIP ?
(26) =(NO THANK YOU)

(27) THANK YOU FOR CALLING.
     GOODBYE.
```

Fig. 1　A transcript of an actual dialog.

る．

本システムは上記の旅行会社の役目をコンピュータに行なわせるように構成した．システムの流れを Fig. 2 に示す．システムは大別すると次の４つの部分に分けることができる．

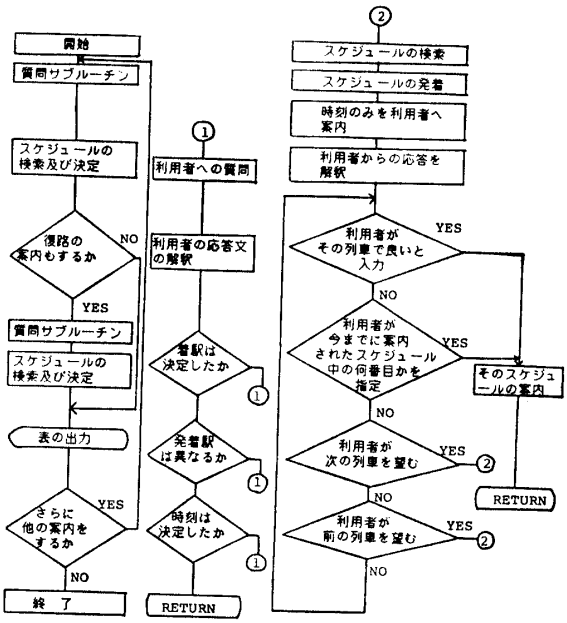
1）システムからの質問

2）客（利用者）からの応答文の解釈

3）スケジュールの計画

4）客へのスケジュールの提示

以下1）～3）の各部分について詳しく述べる．なお4）については Fig. 1(12), (13) のように文章で提示した後，Fig. 1(24) のように表として新めて提示する．



(a) メインルーチン　(b) 質問サブルーチン (c) スケジュールの決定

Fig. 2　Flow charts of program.

3.1　システムの構成

利用する列車を決定する場合，目的地，出発地及び時刻に関する情報の３つの情報を利用者から引き出すために次の３通りの質問を用意した．

Ⓐ　Where do you want to go?　　(目的地)

Ⓑ　Where do you want to leave?　　(出発地)

Ⓒ　What time do you want to leave?　　(時　刻)

システムはⒶ，Ⓑ，Ⓒの順で質問する．しかし質問をする前に質問の目的としている項目（上記質問文の右側のカッコ内の項目）が客によってすでに答えられているような場合はその質問はなされない．例えば，Ⓐの質問で客が

(arrive at Tokyo at 7)

のように応答した場合，応答文には目的地 “at Tokyo” に加えて，時刻に関する情報 “at 7” が含まれているのでⒸの質問はなされない．

本システムでは，本学で使用することを前提に，出発地の初期値として「小郡」が設定してある．「小郡」が目的地である場合以外はⒷの質問はなされない．(Fig. 1(2)～(7))

3.2 客からの応答文の解釈

本システムにおける応答文の解釈では，応答文の主語・述語等を識別し，文全体の意味を完全に解釈しようとするいわゆる自然言語処理的な立場からではなくキーワードに注目した簡便な処理法（キーワード法）を用いている．会話の内容が制限されている場合は，このような方法でも十分であると考えられる．

客からの応答文を解釈する部分はシステム中に2種類存在する．1つは前節で述べたシステム側からの質問に対する客の応答文を解釈する部分，もう1つはシステムが計画をたて，客に提示したとき，その計画に対する客の応答文を解釈する部分である．この節ではこれら2つの部分の応答文の処理について述べる．

3.2.1 システムからの質問に対する応答文の解釈

3.1で述べたシステム側の質問は総て “go” または “leave” という2種類の動詞を使っている．そこで客からの応答文も “go” または “leave”，もしくはこれらの同意語が使われると予想できる．そこで本システムでは，“go”, “leave”, またはそれらの同意語のみが意味のある動詞として扱われる．本システムが意味のある動詞として判断する語を Table 1 に示す．

動詞の他にシステムの中で意味のある語として前置詞と数字がある．

前置詞は “to Tokyo”, “from Kyoto” の “to”, “from” のような場所を表す前置詞と，“before 7”, “after 7”, “about 7”, のような時を表す前置詞を意味のある前置詞と判断する．また，これらの同意語 (Table 2, Table 3) も同様に意味のある語とする．

Table 1 Verbs.

	同　意　語
go	arrive, reach, get
leave	depart, start

Table 2 Propositions of place.

	同　意　語
to	in, at
from	—

Table 3 Propositions of time.

	同　意　語
before	by
after	at
about	—

数字は総て時刻と判断する．例えば “7” は「7時」と判断する．午前・午後はそれぞれ “am”, “pm” とし数字の前後にある場合のみ意味のある語とする．

また，システムは応答文に使用されると考えられる単語，例えば “I”, “want” などの語を用意していて，これらの語は応答文中に存在していても無視する．応答文が，

(I want to go to Tokyo)

の場合，“I”, “want” は無視して “go”, “to Tokyo” という情報のみを取り出す．

以上がシステム中で意味のある語であるが，これら以外の単語は総て駅名と判断する．

さて，具体的な文の処理について解説する．応答文の主なパターンを Table 4, Table 5 に示した．

Table 4 は3.1で述べた各質問文

Ⓐ Where do you want to go?

Ⓑ Where do you want to leave?

Ⓒ What time do you want to leave?

に対して出発地及び目的地に関する情報をどのように処理するかを示したものである．例えばパターン1は質問文Ⓐ, Ⓑ, Ⓒどの場合に対しても

(arrive at Tokyo from Kyoto)

という応答文の場合は

目的地 “Tokyo”, 出発地 “Kyoto”

として処理することを示している．また，パターン4は質問文Ⓐ, Ⓒに対して応答文が

(Tokyo)

であるとき “Tokyo” は目的地を表す語として処理されることを示している．

Table 5 は時刻に関する情報をどのように処理するかを示したものである．この場合はⒶ, Ⓑ, Ⓒのどの質問文に対する応答文であっても同様に処理される．例えば，パターン6は

(arrive at 7)

であるとき，「7時に到着する」という時刻の情報として処理することを示している．

また，応答文が

(arrive at Tokyo from Kyoto at 7)

のように出発地，目的地及び時刻に関する情報を同時に含むときは，Table 4 と Table 5 をあわせて参照すると Table 4 ではパターン1，Table 5 ではパターン6となるから

Table 4 Processing of input sentences (1)

パターン	質問文	動詞	目的地を表す語句	出発地を表す語句	文の特徴
1	A B C	arrive or leave	at Tokyo	from Kyoto	総ての駅名に前置詞をつける.
2	A B C	arrive	Tokyo	(from Kyoto)	動詞があり，駅名がついていない.
3	A B C	leave	(at Tokyo)	Kyoto	〃
4	A C	—	Tokyo	(from Kyoto)	動詞がない.
5	B	—	(at Tokyo)	Kyoto	〃

Table 5 Processing of input sentences (2)

パターン	動詞	到着時刻	出発時刻	文の特徴
6	arrive	at 7		動詞がある.
7	leave		at 7	〃
8	—		at 7	動詞がない.

目的地 "Tokyo", 出発地 "Kyoto"
で「7時に到着する」という意味として処理される.

3.2.2 客がスケジュールの良否を応答する場合の処理

システムは一応のスケジュールを作成すると，スケジュールを客に提示して，客からの応答を待つ．この時の応答文の処理について述べる．処理は次の6種類の応答文について行なう．

① next（又はその同意語）を含む文
② last（又はその同意語）を含む文
③ 序数 (first, second, third, ...) を含む文
④ 時刻に関する情報を含む文
⑤ Yes（又はその同意語）を含む文
⑥ その他の文（No など）

次に上記各文に対するシステムの動作を述べる．

①の文は，現在までにシステムが検索したスケジュールの中で最も早い列車よりも1つ前の列車を検索し，スケジュールを作成する．

②の文は，現在までにシステムが検索したスケジュールの中で最も遅い列車よりも1つ後の列車を検索し，スケジュールを作成する．

③の文は，first が含まれていれば第1番目のスケジュールを客が指定したという意味に解釈し，第1番目に提示したスケジュールについて詳細を出力する．second, third... についても同様である．

④の文は，意味のある動詞と時刻及びその前にある前置詞のみを認識して，その情報にあうようなスケジュールを検索する．但し，ここでは，3.2.1で行なったような目的地及び出発地に関する情報は受け入れず無視する．

⑤の文は，現在提示されているスケジュールについて客が「それで良い」という応答をしたものとして，現在提示されているスケジュールの詳細を出力する．

⑥の文は，客が現在提示されているスケジュールを気に入らないという意味に解釈して
"Do you want a next train or a last train?"
と出力して，再び入力待ちの状態となる．

3.2.3 駅名誤入力の場合の処理

本システムの場合，新幹線の駅名はローマ字で記述することにしている．駅名をローマ字で記述する場合記述方法に個人差がある．例えば

「し」→shi, si「つ」→tsu, tu「ふ」→fu, hu

などの記述法がある．このような様々な記述法は，ある一つの記述を正しい記述として定めれば，その他の記述は誤りの記述として扱うことができる．つまりローマ字の記述の個人差の処理は例えば "Kokura"（小倉）の u と e を誤って 'Kokera" としたときこれを "Kokura" と記述しようとしたものであろうとして処理する場合と同様に処理することができる．

そこで具体的に誤り記述の解決策の1つとして誤り数という考えを用いた．誤り数とは誤り記述に文字の代入，挿入又は削除を行なって正しい記述にするとき

の文字数の最小値である．次に例をあげて説明する．

（例1）正しい記述 abcd, 誤り記述 aecd とすると，誤り記述の e に b を代入すると abcd となる．従って誤り数1となる．

（例2）正しい記述 abcd, 誤り記述 abd の場合，c を挿入すれば正しい記述となる．従って誤り数1となる．

（例3）正しい記述 abc, 誤り記述 adbc の場合，d を削除すれば正しい記述となる．従って誤り数1となる．

ここで（例2）の場合，誤り記述の d に c を代入して (abd→abc), さらに d を挿入 (abc→abcd) しても正しい記述となるが，誤り数は2となるから最小値をとって誤り数1とする．

このような誤り数を入力された駅名と総ての駅名に対して求めて誤り数の最小である駅名を正しい駅名と判断する．

ところが，札幌などのように新幹線には全く関係のない駅名が入力された場合にも誤り数の最小の駅名を探すことはできる．しかし，このような場合には札幌は「新幹線の駅名にはない」というような判断が下されるべきである．そこで誤り数に閾値を設け，閾値以上の場合には入力駅名は「新幹線の駅名にはない」と判断することにする．実際には本システムでは誤り数の閾値を入力文字数の1/3（小数部切り捨て）とした．

又，駅名を省略して入力する場合も考慮に入れた．例えば正しい記述 abcde に対して abc は，abc の後に de を挿入すれば abcde となる．従って本来ならば誤り数2であるが，誤り記述の後に挿入した文字数は誤り数には加えないこととした．従って前例 abcde に対して abc は誤り数0となる．

さらに，特に新幹線の駅名には「新」のつく駅名が多い．例えば新下関，新大阪などである．本システムでは入力駅名が「新幹線の駅名にはない」という判断が下された場合，入力駅名に “Shin-”（新）を付加して再び処理することにした．例えば “Osaka” と入力した場合，「新幹線の駅名にはない」という判断が下される (Shinosaka と比較すると誤り数4であり閾値1より大きい) から “Shin-” を付加して “Shinosaka” として処理する．

以上の処理により，キーボードのうっかりした押し誤りや，前述したローマ字の記述方法の個人差などが大幅に解消できた．

しかし誤り数が最小である駅名が2つ以上ある場合例えば Tokyoh と入力した場合，Tokyo, Toyohashi ともに誤り数1である．このような場合，本システムでは入力者（利用者）にどれかを選択してもらうことにした．(Fig. 1(3), (4) 参照)

3.3 スケジュールの計画

3.1で述べた3つの情報（目的地，出発地及び時刻に関する情報）がそろうとスケジュールを計画することができる．以下，本システムのスケジュールの計画の方法について述べる．

まず新幹線の駅の中で，最も停車駅の少ない列車の停まる駅を乗換え駅とする．乗換え駅は

東京，名古屋，京都，新大阪，岡山，広島，小倉，博多

の各駅である．ここでスケジュールを作成する場合，時刻に関する情報で「出発地をある時刻に出発する」場合と，「目的地にある時刻に到着する」場合があるが，出発地から検索するか目的地から検索するかの違いだけで，検索法は全く同じである．

出発地から検索する場合を考える．典型的な場合は出発駅と目的駅の間に乗換え駅が2つ以上存在する場合である．このとき出発駅をA駅，目的駅をB駅，出発駅から目的駅の方向へ最も近い乗換え駅をC駅，目的駅から出発駅の方向へ最も近い乗換え駅をD駅とする．つまり駅はA，C，D，Bの順に並んでいる．(Fig. 3(a))

次に検索法を4つの場合に分けて説明する．

① A，B両駅が乗換え駅である場合は直通列車を探す．(Fig. 3(b)) また，A，B両駅間に乗換え駅がない場合にも直通列車を探す．

② A駅が乗換え駅であって，B駅が乗換え駅ではない場合，D駅が存在しなければ①の場合に帰着する．D駅が存在すればA，D駅直通列車を探した後，D駅着時刻を新たにD駅発時刻としてD，C駅間のスケジュールを探す (Fig. 3(c)).

③ A駅が乗換え駅ではなく，B駅が乗換え駅である場合，②のD駅がC駅にかわっただけで②と同様に考えられる (Fig. 3(d)).

④ A，B両駅ともに乗換え駅ではない場合，A，

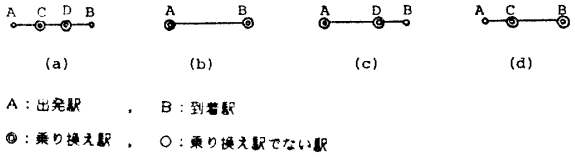


Fig. 3 Patterns between departure station and arrival station.

C駅間直通列車を探し，その列車のC駅着時刻を新たにC駅発時刻とし，C，B駅間のスケジュールを②の場合に適用して計画する (Fig. 3 (a)).

4. むすび

LISP を用いて試作した対話型の新幹線列車案内システムについて述べた．

新幹線の列車を選択するときには，時刻表を参照し目的に合った列車を見つける．本システムは時刻表を参照するというわずらわしさを解消した．また，応答文に対して簡易な言語処理を行なうため，システムの使用法を知らなくても，起動法さえわかればシステム側の質問に答えていくことによって列車を選択することができる．

残された問題として乗換えルールの変更と応答文中の駅名の判断がある．

まず，乗換えルールの変更であるが，本システムでは3.3節で述べたように，乗換え駅があれば必ず乗換える方法をとっている．乗換えルールは他に，目的駅までの直通列車を探す法，最も早く目的地へ到着する計画をたてる法（例えば，小郡から新横浜へ行く場合東京まで行き，東京から新横浜まで戻った方が早い場合がある.）などが考えられる．これらのルールを取り入れることができればより利用者の注文にあうようなスケジュールを計画することができる．

応答文中の駅名の判断の問題であるが，本システムでは3.2節で述べたように，システム中で意味のある語やシステム中にあらかじめ用意された語以外は総て駅名と判断する．従って全く関係のない語が入力された場合にも駅名と判断することになる．この問題の解決には，より多くの応答例を考慮してシステム中に用意する単語の数を増すことが必要である．

本システムは列車を東海道・山陽新幹線に限ったがデータファイルの追加及び数個の関数の変更により在来線との接続及び他の列車案内システム・旅行案内システム等に応用することが容易に可能である．

謝辞　日頃，ご助言頂く本学高浪五男教授，井上克司助教授，ならびに LISP システムに関して有益なるご意見を頂いた本学石原好宏助教授に感謝する．又，本研究に際し有益なるご意見を頂いた本学谷口弘氏に感謝する．

文　献

(1) Bobrow, D. G. and Collins, A.M.：“人工知能の基礎；知識の表現と理解”，淵一博訳，近代科学社 (1978)
(2) Daniel G.Bobrow, Ronald M.Kaplan, Martin Kay, Donald A. Norman, Henry Tompson, Terry Winoglad：“Gus. A. Frame-Driven Dialog System”, Artifficial Intelligence. 8, 155-163 (1977)
(3) Laurent Siklóssy：“LISP入門”，後前英一・戸島淵訳，日本コンピュータ協会
(4) 中西正和：“LISP 入門’，近代科学社 (1981)
(5) “ACOS-6 プログラム管理 LISP1.5 プログラミング説明書”，日本電気 (1981)
(6) 斉藤正男，溝口文雄：“知的情報処理の設計”，コロナ社 (1982)
(7) 溝口文雄，北沢克明：“知識工学入門”，講談社 (1982)

(昭和59年4月14日受理)