

ミラー効果に基づく相互チュートリアルによる 複数インタフェースでの操作方法学習支援

桂川 景子^{†1,†2} 大野 健^{†1} 富 樫 実^{†1}
小 暮 悟^{†2} 伊 藤 敏 彦^{†3}
小 西 達 裕^{†2} 伊 東 幸 宏^{†2}

近年、多くのユーザが PC や携帯電話、カーナビなど複数種類の情報端末機器を所有し、利用するようになってきた。これにともない、同一のサービスを環境に応じてインタフェースを変更しながら利用可能な仕組みへの要求も高まっている。しかし複数種類の端末機器を使用する際には、同一のシステムであっても各機器に備わるインタフェースでの操作方法を個別に習得しなければならず、扱える端末機器が増えるほどユーザの負担は増大する。本論文ではこの問題を、ミラー効果に基づく相互チュートリアル機能を各インタフェースに持たせることで解決する。ミラー効果に基づく相互チュートリアルとは次の 2 つの特徴を持つ操作方法の学習支援機構である。(1) 同一システム上の同一タスクを実行する際には、どのインタフェースを使用する際であっても、同一のプログラムと入力データが利用されることに着目し、異なるインタフェース上で同一の操作結果が得られる操作手順を生成する。(2) ユーザがあるインタフェースを利用した際に、同一タスクを他のインタフェースで実行するための操作手順を生成・デモンストレーションすることで他のインタフェースでの操作方法を自然に身につけさせる。カーナビでのメニュー操作インタフェースから PC での自然言語インタフェースへの相互チュートリアル機能を実装したシステムの評価では、タスク達成時間が約 24%短縮、入力文の受理率は約 17%向上し、その有効性が示された。

A Mirror-effect-based Mutual Tutorial System for Learning Operations on Different Interfaces of the Same Software Service

KEIKO KATSURAGAWA,^{†1,†2} TAKESHI OONO,^{†1}
MINORU TOMIKASHI,^{†1} SATORU KOGURE,^{†2}
TOSHIHIKO ITOH,^{†3} TATSUHIRO KONISHI^{†2}
and YUKIHIRO ITOH^{†2}

Recently, cell phones, PCs and car navigation systems are increasingly used for taking advantage of a single software service. Although such a system typically offers different interfaces according to the users' environments, not every user is familiar to the operation of all the available devices; hence, the users face difficulties in switching from one device to another. One of the biggest problems of a service accessible from multiple devices is that the users must learn different operations on different interfaces. Needless to say, this is a heavy burden on the users and it is desirable to alleviate the load. To solve this problem, we propose a mirror-effect-based mutual tutorial system to support learning operations on different interfaces. The basic functions of our tutorial system involve the following two procedures. 1) By focusing on a pair of a program and its input data for conducting a task, the system generates operation procedure to perform the same task on different interfaces. 2) By demonstrating the generated procedures, the system helps the users learn operations on different interfaces. It is experimentally confirmed that the tutorial system improves the usability. The task completion time is reduced by 24% and the input acceptance rate is increased by 17%.

1. はじめに

今日、我々の周りには多くの情報提供サービスが存在するとともに、情報端末機器の種類や、各サービスを受けるユーザの環境も多様化している。また、これらのサービスを利用するためのインタフェースも実に様々なものがある。

たとえば、多くのカーナビゲーションシステム（以下カーナビ）には、リモコンやスイッチに加え、運転中の操作のために音声入力マイクが備えられており、状況に応じて入力に使用するデバイスを変更可能となっている。また、カーナビからのインターネット接続が可能になったことにより¹⁾、PC や携帯電話から主に利用されていたインターネットサイトが車室内からも利用可能となった。そのため、多くの情報提供システムで、異なる複数の端末機器からアクセスされることを前提とすべき状況となってきた。同一の情報提供システムに対して、異なる複数の端末機器からアクセス可能なサービスの例としては、ユーザが携帯電話と PC および車載器を状況に応じて使い分けて利用することができるグルメ情報検索サイトのぐるなび²⁾などがあげられる。

^{†1} 日産自動車株式会社
NISSAN MOTOR CO., LTD.
^{†2} 静岡大学
Shizuoka University
^{†3} 北海道大学
Hokkaido University

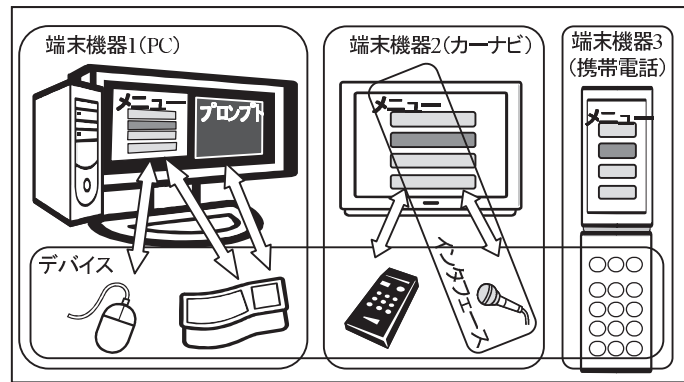


図 1 複数インタフェースの例
Fig. 1 Example of plural interfaces.

図 1 に複数インタフェースを持つシステムの例を示す。ここでは、PC やカーナビなどの端末機器に備えられているマウスやマイクなどの入力装置をデバイス、メニュー選択やコマンド入力などの入力方式をインタラクションスタイルと呼ぶ。また、各端末機器が備えるデバイスとインタラクションスタイルを組み合わせた入出力機構をインタフェースと呼ぶ。

こういった複数のインタフェースを備えるシステムは数多く存在するが、複数のインタフェースを状況に応じて使い分けることを想定したインタフェース設計方針を示す例はほとんどなく、多くの場合、各インタフェースは個別に設計されている。よってユーザは、あるインタフェースからの操作に慣れたシステムであっても、未経験のインタフェースで操作する際には、初めてそのシステムを操作したときと同様の試行錯誤を繰り返しながら、操作方法を習得する必要がある。このため、システムにアクセス可能なインタフェースの種類を増やしても、個々の操作性を向上させるだけでは、ユーザビリティの向上に直接つながりにくい。

複数インタフェースを使い分ける際のユーザの負荷は、様々な情報端末を状況に応じて使い分けるユビキタス環境を想定したシステムでは避けては通れない大きな問題となる。このように、ユーザが状況に応じてスムーズに使い分けられることができるインタフェースへの要求は次第に高まってきている。

特に車載器においては、運転中と停車時でユーザの状況が大きく異なるため、それぞれの状況に応じたインタフェースを個別に提供しなければ、ユーザビリティは著しく低下する。

たとえば、運転中には安全上の制約からリモコンやタッチパネルからの操作は制限されているし、仮に操作可能としたとしても、運転をしながら速やかに操作することは困難である。この場合、音声入力インタフェースであれば、前方から目をそらすこともハンドルから手を離すこともなく操作できるため、容易に操作可能である。一方、停車時にはリモコンやタッチパネルからの手操作による入力も、音声入力も可能であるが、複数人の同乗者が会話していたり、音楽が流れていたりする場合など、音声入力よりもリモコンやタッチパネルからの入力の方が確実な場合もある。また、PC や携帯電話、車載器からのアクセスを想定するインターネット上のシステムは、各インタフェースの使用環境が大きく異なるため、個別に使いやすいインタフェースが提供される必要がある。よって、複数インタフェースをスムーズに使い分けられることができる仕組みは、特に車載器からの利用を想定したシステムにおいて非常に重要となる。

そこで本論文は、同一の機能^{*1}を異なるインタフェースで利用可能なシステムにおいて、インタフェースごとに異なる操作方法^{*2}を個別に習得する負担を軽減することを目的とする。一般に、同一のシステムにおいてある端末機器からある機能を利用するために高い頻度で用いられる戦略^{*3}は、ある程度の有効性を経験的に知っている戦略であり、他の端末機器で同じ機能を利用する際にも用いられる可能性が高いと考えることができる。我々はこの点に着目して、ミラー効果に基づく相互チュートリアル機能を開発した。

ミラー効果に基づく相互チュートリアル機能とは、インタフェースごとに異なる操作方法を個別に習得する負担を軽減することを目的とする学習支援機構である。ミラー効果に基づく相互チュートリアル機能は、ユーザがあるインタフェースを使用してシステムを操作した際に、他のインタフェースで同一の機能を同一の戦略で利用する際の操作方法を提示することで自然に別のインタフェースでの操作方法を習得させる。これによって、同一のシステムを複数のインタフェースで操作できることの利点を生かした情報機器の操作環境構築を

*1 ここでは、カーナビにおける施設検索や経路編集、列車予約システムにおける列車予約や予約確認など、ユーザの目的を実現する作用を「機能」と呼ぶ。

*2 施設検索を行うためのメニュー選択型のインタフェースにおいて、たとえば「浜松市のガソリンスタンド」を検索するというタスクを実行するために、県名メニューから「静岡県」を選択してから、市区町村メニューから「浜松市」を選択し、施設ジャンルメニューから「ガソリンスタンド」を選択するなどといった操作の手順と実際の入力内容を「操作方法」と呼ぶ。

*3 たとえば、施設検索機能を利用する際には、市区町村名と施設ジャンルを指定して検索するという方法、施設名を直接指定するという方法など、複数の方法をとらう。列車予約機能の利用時にも、出発時刻を指定して列車を選択して予約する方法、割引料金が適用される列車を選択して予約する方法などがある。ここでは、ある機能を利用するためにとらう個々の実現方法を「戦略」と呼ぶ。

指す。

また、我々がこれまでに構築したドライブプランニングシステム^{3),4)} (以下 DP システム) を例として、PC 上の自然言語インタフェースとカーナビゲーションシステムのメニュー操作インタフェース間での相互チュートリアル機能の実装とその評価実験について述べる。

2. 本論文のスコープ

本章では、複数のインタフェースを備えたシステムのユーザインタフェース設計における課題を整理し、我々の研究スコープを明らかにする。

対象とするシステムは同一機能が複数のインタフェースから同様に頻繁に利用されるものとする。また、対象とするユーザは一方のインタフェースに慣れているが、他方のインタフェースには慣れていないユーザを想定する。これらのスコープは、「同一のシステムにおいて、ある端末機器からある機能を実行するために高い頻度で用いられる戦略は他の端末機器からも利用される可能性が高い」という仮定に基づいて設定した。また、慣れていないインタフェースであっても、マウスの操作方法やキーボードからの日本語入力方法など、入力デバイスの基本的な使い方については習得されているものとする。

この範囲内で、さらに我々の研究スコープを以下のように設定する。

- 向上させるユーザビリティ要件：効率と学習しやすさ
- 評価基準：受理率とタスク達成時間
- 対象とするインタラクションスタイル：自然言語インタフェースとメニュー選択インタフェース
- 学習支援内容：あるインタフェースで使用された機能を別のインタフェースから同じ戦略で実行するための入力手順と具体的な入力内容

以下、本章ではこれらのスコープについて詳しく述べる。

2.1 本論文が対象とするユーザビリティ要件と評価基準

Nielsen⁵⁾ はユーザインタフェースのユーザビリティとは、学習しやすさ、効率性、記憶しやすさ、エラー発生率、主観的満足度の 5 つの特性によって構成されるものと定義している。

これまでもこれらの特性を改善する取り組みは数多くなされてきた。機器操作におけるユーザのメンタルモデル形成過程を明らかにする研究⁶⁾などは学習しやすさや記憶しやすさの向上に貢献している。また、Unix システムにおけるエイリアスの設定などは操作の簡略化による効率の向上例である。しかしこれらは、主に単独のインタフェースにおけるユー

ザビリティ向上のみを目的としている。

そこで我々は、複数のインタフェースを備えたシステムにおいて、少なくとも 1 つのインタフェースからの操作に慣れたユーザが、使い慣れていないインタフェースから操作する際のユーザビリティ向上を目指す。

特に本論文では Nielsen の定義による 5 つのユーザビリティ特性のうち、学習しやすさと効率性に着目する。ISO9241-11⁷⁾ では効率性を「ユーザが目標を達成する際に、正確さと完全性に費やした資源」と定義している。ここでは、資源にあたるもののうち、時間をとりあげ、正しい入力操作を短い操作時間で実行できる操作方を効率性の高い操作方と定義する。また、効率性の高い操作方を強い学習意識や特別な学習時間を設けることなく自然に身につけられることを学習のしやすさと定義し、ユーザ入力の受理率、タスク達成時間によって提案手法を適用したシステムのユーザビリティを評価する。

2.2 対象とするインタラクションスタイルと学習支援内容

Shneiderman⁸⁾ は、対話型ソフトウェアにおける代表的なインタラクションスタイルとして、メニュー選択、空欄記入、コマンド言語、自然言語、直接操作の 5 つをあげている。本論文ではこれらのうち、PC、携帯電話など多くの端末機器で共通に用いられるメニュー選択および自然言語形式を取り上げる。

また Foley ら⁹⁾ は、グラフィカルインタフェースにおける操作の最小単位である基本対話タスク (Basic Interaction Task: BIT) を、テキスト入力、選択、位置指定、量化の 4 つに分類している。テキスト入力、メニューからの選択などの BIT を複数回繰り返して操作を行う場合、BIT の順序集合 (BIT 列) がタスク達成に必要な操作手順となる。ユーザがシステム上でタスクを実行する作業は、インタラクションスタイルおよびデバイスを選択して、BIT または BIT 列を実行する作業であるといえる。よってインタラクションスタイルとデバイス、これらの組合せで実行される BIT および BIT 列、その結果実行されるタスクの関係は図 2 のようにとらえることができる。

図中、インタラクションスタイルとデバイスを結ぶ線は、選択可能な組合せを表している。たとえば、自然言語によるインタラクションスタイルは、キーボードもしくはマイクと組合せが可能である。デバイスと実行 BIT 列を結ぶ線は、各デバイスで主に使用される BIT の種類を表している。たとえばキーボードはテキスト入力と選択によく使用される。実行 BIT 列の枠内にある文字列 (T1, S1 など) は実入力データを表している。それらとタスクを結ぶ線は、タスクとその実行に必要な入力の関係を示しており、各タスクは結ばれたすべての入力が行われた際に実行される。たとえばタスク F4 は選択 S1+S2+S4 によって

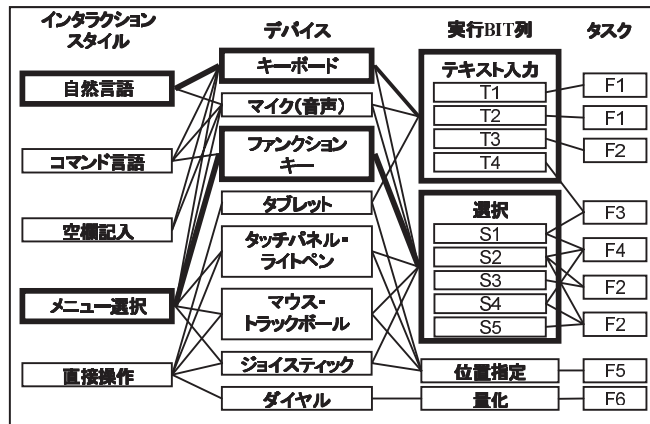


図2 インタラクションスタイル・デバイス・BIT列・タスクの関係
Fig.2 Relation with interaction styles, devices, BIT sets and tasks.

実行される。

状況に最も適したインタラクションスタイルやデバイスを選択することも、複数インタフェースを持つシステムを使用する際の分かりにくさとなりうるが、本論文では、使用可能なインタフェースが決定している場合の、インタフェースの使い方を学習する段階に焦点を当てている。

インタラクションスタイルおよびデバイスが決定している場合、テキスト入力や選択といった、実行すべき BIT の種類はほぼ決定する。しかし各 BIT で指定する情報およびその入力順序は、タスクおよびインタフェースごとに学習する必要がある。特にメニュー選択では 1 度の BIT で入力可能な情報量が少ないため、複数回の操作によってのみ実現可能な戦略が多く、その組合せが複雑で分かりにくくなりやすい。

さらに、同一タスクを実行する BIT の組合せは複数ありうる。たとえば図 2 でタスク F2 を実行する BIT 列には、T3, S2+S3, S2+S4+S5 の 3 通りがある。このように、BIT の種類によって必要な操作回数が異なったり、同種の BIT であってもその組合せによって操作効率が異なったりする。これも操作方法の学習を困難にしている要因の 1 つと考えられる。

また、個々の BIT を実行する際には具体的な入力内容を決定する必要があるが、自然言語入力のように同じ意味内容を指定する表現が複数ある場合がある。図 2 のタスク F1 はテキスト入力 T1, T2 のどちらでも実行可能となっている。ただし、現状の自然言語インタ

フェースでは受理できる表現が制限されていることが多く、こういった表現で入力すれば受理されるかが分からないことも多い。このため、どのような単語・文体を用いて、入力する自然言語文を構成すればよいかを学習する必要が生じる。

そこで我々は、異なるインタラクションスタイルとデバイスの組で構成される複数のインタフェースを持つシステムにおいて、効率的な BIT の入力手順および具体的な入力表現をインタフェースごとに学習する煩雑さの軽減を目指す。本論文では、図 2 中、太線で示した、キーボードによる自然言語インタフェースと、ファンクションキーを用いたメニュー選択インタフェースを持つシステムを取り上げ、効率的な入力手順と入力表現の学習を支援する方法を提案する。

3. ミラー効果に基づく相互チュートリアル

我々は、複数のインタフェースを持つシステムにおけるユーザビリティ向上手段として、ミラー効果に基づく相互チュートリアルという手法を提案する。本章では、この手法について詳しく述べる。

3.1 相互チュートリアルの定義

相互チュートリアルはユーザが実際に実行したタスクごとに、同一のタスクを他のインタフェースで効率的に操作する方法をデモンストレーションする。これによって、あるインタフェースを利用して行った操作を他のインタフェース上で効率的に実行する方法が鏡像のように自動的にユーザに提示され、ユーザに積極的な学習行動をとらせることなく、自然に他のインタフェースの利用法を習得させることができる。

特定の機能を利用するために、あるインタフェースからの使用頻度が高い戦略は、ユーザがその戦略の有用性を知っているため、他のインタフェースからも使用される可能性が高いと考えられる。ミラー効果に基づく相互チュートリアルでは、あるインタフェースで実際に利用された戦略についてのみ、他のインタフェースでの操作方法を学習させることで、ユーザにとって必要な戦略の利用方法に絞って効率的に学習させることができる。これにより、新たに操作方法を習得したインタフェースにおける入力の受理率向上が期待できる。

あるインタフェースで利用された機能を同一の戦略で他のインタフェースから利用しようとする場合、他のインタフェースからは利用できない場合や、複数通りの操作方法で利用可能な場合もある。後者の場合には、それらの操作方法の間に効率の差がある場合もある。ミラー効果に基づく相互チュートリアルでは、同じタスクを同じ戦略で利用できない場合には、その旨を教示する。また、複数通りの操作方法がある場合には、それらのうち、より効

率的な操作方法をチュートリアルする．これにより，タスク達成時間の短縮が期待できる．

3.2 他のインタフェースでの効率的操作方法生成

図3に各インタフェースで行われたタスク t と同じタスク t を他のインタフェースで実現する効率的な操作方法を生成するシステムの構成を，メニューと自然言語の場合を例にあげて示す．

自然言語 BITsetX およびメニュー BITsetX は，それぞれあるタスク t を実行するためにシステムの機能 f をある戦略 s で利用するための操作方法のうち，ユーザが実際に実行した操作を表す．

ユーザ入力時におけるインタフェースの役割は，インタフェースからの入力を解釈して，必要なプログラムを特定し，そのプログラムへの入力データを生成することである．そのため，各インタフェースにはユーザからの入力を解釈するインタプリタの役割を果たす機能が備えられている．図3では，インタプリタ N ， M がそれぞれ自然言語入力，メニュー選択入力を解釈するインタプリタを示している．

自然言語 BITsetX およびメニュー BITsetX は，それぞれインタプリタ N ， M によって解釈される．この解釈により，機能 f を利用するためのプログラムが特定されて，入力デー

タが生成される．通常，ユーザ操作 (BITsetX) からの実行プログラム列の特定および各プログラムへの入力データの生成は一意に行われる．また，各インタフェースからの入力が，同一機能 f 利用した同一戦略 s による同一タスク t の実行を指示している場合，ユーザが選択した操作 (自然言語 BITsetX またはメニュー BITsetX) が冗長であったとしても，システム内部では，それぞれの入力は同じプログラム列と同じ入力データの組合せとなる．

ここで，実行プログラム列およびプログラム列への入力データから各インタフェースにおけるユーザ操作方法を生成する逆インタプリタをそれぞれ用意することができれば，あるインタフェースの入力から任意のインタフェースでの操作方法を生成できる．図3では，インタプリタ N^{-1} が自然言語入力による操作方法を生成する逆インタプリタ， M^{-1} がメニュー選択による操作方法を生成する逆インタプリタとなっている．

逆インタプリタを実装する際には，実行プログラムおよびプログラムへの入力データに対応する操作方法が1つも存在しない場合，1つだけ存在する場合，複数存在する場合の3通りを考慮する必要がある．

実行プログラムおよびプログラムへの入力データから，生成可能な操作方法が存在しない場合，そのインタフェースではこれらの実行プログラムおよび入力データで実行されるタスクが操作できないということになる．この場合は，そのインタフェースでは実行不可能であることを示すなどすればよい．

操作方法が1つだけ存在する場合は，これを逆インタプリタ結果として選択する．この場合，この操作方法がチュートリアルのために提示すべき操作方法となる．図中ではこれを自然言語 BITsetMin あるいはメニュー BITsetMin で示している．

複数存在する場合は，生成しうる操作方法のうち，操作時間が短いと見込まれる操作方法を選択する．これによって，効率的な操作方法を学習させることができる．なお，操作時間の短い操作方法の選択ロジックはインタフェースごとに設計する必要がある．見込まれる操作時間の間に大きな差がない場合 (たとえば，自然言語インタフェースである文を入力するという方法と，その中の単語を同義語に置き換えた別の文を入力するという方法など) は，あらかじめ決めた代表的な操作方法を選択する．図3の，自然言語 BITsetMin は，ユーザがメニュー選択インタフェースから戦略 s により行ったタスク t を自然言語インタフェース上で，同じく戦略 s により実行する際の効率的な操作方法 (もしくは代表的と定められた操作方法) を表している．同じくメニュー BITsetMin は，自然言語インタフェースから戦略 s で行ったタスク t をメニューインタフェース上で同じく戦略 s で実行する効率的な操作方法である．

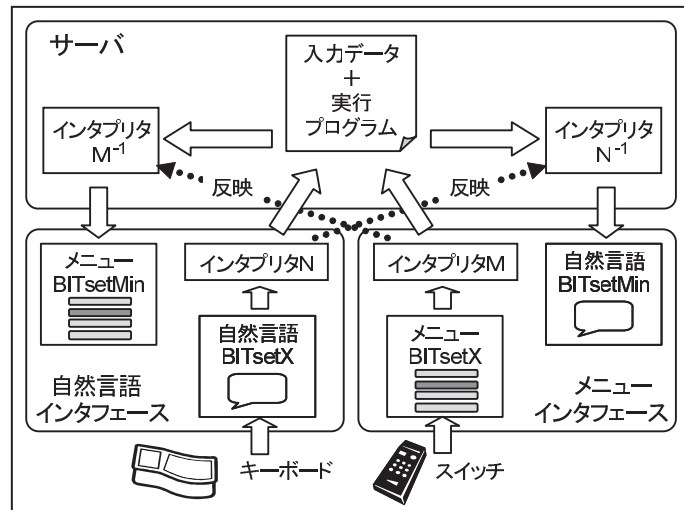


図3 他のインタフェースでの操作方法生成
Fig.3 Generating operation for another interface.

このように、効率的な操作方法を選択する逆インタプリタを用意することで、ユーザが一方のインタフェースで効率の悪い方法で操作を行った場合（自然言語 BITsetX, メニュー BITsetX が冗長性を持つ場合）にも、各々で最も効率的な操作方法を示すことができる。

図3では、自然言語入力またはメニュー選択という1種類のインタラクションスタイルによってタスクを実行する際の例を示したが、複数種類のインタラクションスタイルの組合せによってタスクを実行するような場合でも、同様の方法で相互チュートリアル可能である。

4章では、メニュー選択を用いたインタフェースと自然言語入力を用いたインタフェースでの逆インタプリタの設計例および、これを利用したミラー効果に基づく相互チュートリアル具体例について説明する。

4. DPシステムにおける相互チュートリアル

4.1 DPシステム

今回我々は、複数のインタフェースから同一の戦略で同じ機能を利用可能なシステムの例として、DPシステム³⁾を取り上げ、相互チュートリアル機能を実装、評価した。DPシステムとは、PC上で作成したドライブのプランデータをオンラインサーバ上にアップロードし、カーナビや携帯電話での経路案内や施設検索、プラン編集などに役立てるシステムである。

本章では、PCとカーナビ上のインタフェースで共通に利用可能な施設検索機能における相互チュートリアルについて述べる。

PCでは、その操作自由度を生かして、キーボードによる自然言語入力インタフェースを採用している。この自然言語入力インタフェースは、ユーザがその要求を思いついたまま入力可能とすることを目指しており、「浜松城の近くでホテルを探して」などという、複雑な条件も1度に指定することができる。カーナビ上のインタフェースでは、スイッチ操作によるメニュー選択形式を採用している。カーナビでも、近隣施設を指定して訪問地を検索するといった比較的複雑な操作も受理可能であり、PCシステムで利用可能な、様々な訪問地検索機能の大部分が利用可能となっている。

4.2 DPシステムにおける他インタフェースでの効率的な操作方法生成

本システムの施設検索モジュールでは、施設のカテゴリ、住所、近隣施設とそこからの距離や時間を検索の条件として指定可能となっている。指定された検索条件は木構造で表現され、検索モジュールに渡される。DPシステムの検索モジュールおよびそれに渡される木構造は、DPシステムの施設検索機能を利用するために専用に設計されている。木構造は、施

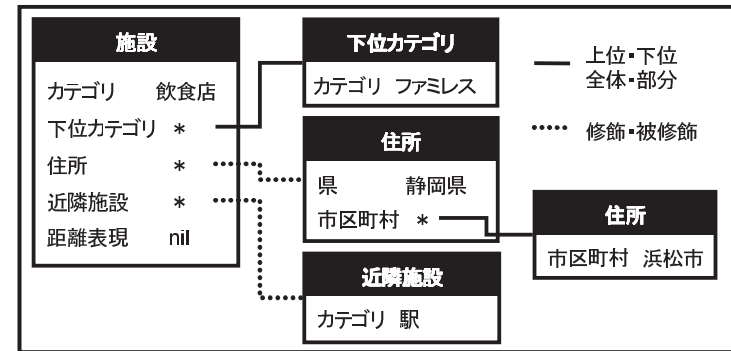


図4 検索用木構造の例
Fig. 4 Tree structure example.

設を表す施設フレームと検索の条件を表すフレームおよびこれらのリンクとで表される。

図4に「静岡県浜松市で、駅の近くのファミレス」を検索する木構造を示す。木のノード間は、概念間の上位下位階層または全体部分階層を表すリンク（図中の実線リンク）と、概念間の修飾・被修飾関係を表すリンク（図中の点線リンク）とで接続されている。木のノードはフレームで構成され、属性名と、属性値あるいは属性値へのポインタのペアを格納する。たとえば、住所の全体フレームには、属性名“県”とその属性値“静岡県”，属性名“市区町村”と属性値である住所の部分フレームへのポインタが格納されている。

「駅の近くの」、「現在地の近くの」のように近隣施設を指定する条件の場合、基準点となる施設フレーム（「駅」や「現在地」を現す施設フレーム）が近隣施設スロットに格納される。この場合、距離表現スロットは「nil」となり、検索モジュールは近隣施設で指定された施設から5km以内というデフォルト値を利用して検索する。また、「浜松城から10km以内の」、「現在地から30分以内の」といった、基準点となる施設と距離または時間によって与えられる検索条件は、近隣施設スロットに基準となる施設の情報と格納し、距離表現スロットに「1km以内」「30分以内」などの距離または時間表現を格納することで表現している。

なお、どのインタフェースからであっても、同一のタスクを同一の戦略で実行する際に使用する、DPシステムの検索モジュールおよびそれに渡すために生成される木構造は同一のものとなる。したがって、この検索モジュールおよび木構造を、3章で説明した共通の実行プログラムおよびプログラムへの入力データを指定するものとして、あるインタフェースでの操作方法を逆生成するためのソースと位置づけることができる。

DP システムの自然言語入力結果から木構造への変換方法は、文献 3) の方法を用いている。メニュー選択入力結果から木構造への変換は、メニューあるいはそのメニュー項目ごとに木構造の属性との対応を定義し、メニューから選択されたメニュー項目を木構造の該当する属性に伝播させるという方法をとっている。以下では、それぞれのインタフェースにおける、木構造から操作方法への逆変換の手順を示す。

4.2.1 木構造からの自然言語入力逆生成

木構造から自然言語入力文を生成する場合、まず初めに各フレームに納められた属性値に相当する表層表現を抽出する。上位下位（あるいは全体・部分）関係で階層化された施設カテゴリおよび住所は、上位（全体）の表現と下位（部分）の表現を“の”で接続する。

ただし、上位（全体）階層による限定がなくても下位（部分）階層の表現のみで指示する内容にあいまい性がなく、対象を一意に指定できる場合には、効率的な自然言語表現を生成するために、上位（全体）の表現を省略し下位（部分）の表現だけを用いる。たとえば、部分表現である市区町村名の「朝日町」は、2008年6月現在、山形県、長野県、三重県の3県内に存在する。そのため、これを自然言語で指定する際に、全体表現である県名を指定しないで市区町村名を指定するとあいまい性が残る。一方、「横浜市」のように同一の市名が存在しない部分表現は、全体表現である県名（神奈川県）を省略してもあいまい性がなく一意に市を特定することができるため、県名を省略した表現を選択する。

次に、図4で施設フレームに点線のリンクで接続している修飾情報（住所フレームおよび施設フレーム）から変換される表層表現を、被修飾情報を表す表層表現（木のルートの施設フレームとその下位カテゴリスロットからリンクされるフレームから変換される）に連絡語（“の”“で”“の近くの”など）を介して結び付けることにより、自然言語表現を生成することができる。連絡語および複数の修飾情報を持つ場合の接続順序は、フレームの属性ごとにあらかじめ決めておく。なお修飾情報が被修飾名詞の直前に置かれる場合と、被修飾名詞との間に別の修飾情報が挿入される場合で別の連絡語を用意することができるようにしている。

たとえば、メニュー操作の結果図4の木構造が生成された場合、“飲食店”と“ファミレス”および“静岡県”と“浜松市”はそれぞれ階層化された概念であるが、最下層の語だけでもあいまい性がない。よって、最下層の“ファミレス”と“浜松市”をそれぞれ効率的な入力表現として選択する。修飾情報である住所と近隣施設では住所を先に指定するよう定めている。また、住所・近隣施設の順で2つの修飾情報を指定する場合の被修飾語との連絡語は、それぞれ“で”、“の近くの”と定めている。よって、「浜松市で駅の近くのファミレス」が自然言語入力における効率的な入力方法として生成される。

4.2.2 木構造からのメニュー操作方法逆生成

メニュー選択インタフェースのメニューおよびその項目は施設検索用の木構造に検索条件をあてはめるために設計されており、各メニューで選択された情報が木構造の各フレームで表される属性のうち、どの属性を指定するものかはあらかじめ一意に対応付けられている。よって、木構造中に属性値が格納されている属性ごとにこの逆をたどれば、どのメニューからどのメニュー項目を選択すればよいかを決定でき、メニュー選択操作を生成することができる。

ここで、階層化された属性値の下位（部分）階層の指定は上位（全体）が指定された場合にのみ可能であるため、上位（全体）の指定を先に行う。それ以外の属性値指定順序は、基本的には任意であるが、視線の移動方向に沿って、属性値を選択させるほうが自然であるため、メニュー選択インタフェースにおいてメニュー上に配置されるメニュー項目の順序に沿って、上から下に順次指定させる。

複数の操作方法で同一の値を指定可能な場合は、操作回数の少ない方を選択する。DPシステムにおけるメニュー構成では各メニューの選択に要する時間には大きな差がなく、操作回数で効率を近似することができるためである。

図4に示した木構造からメニュー選択手順を生成する例を示す。「静岡県」と「浜松市」は全体部分関係にあるため、“静岡県”を選択した後、“浜松市”を選択させる。施設の上位カテゴリである“飲食店”と“ファミレス”は上位下位関係にあるため、“飲食店”を先に選択させる。施設カテゴリと住所、近隣施設である“駅”は、どの条件からでも選択可能であるが、施設カテゴリ、住所、近隣施設の順で画面上に表示されるため、この順序での選択を採用する。

4.3 DPシステムにおける相互チュートリアル

DPシステムにおいて、PC上の自然言語入力インタフェースからの、カーナビのメニュー選択インタフェースへのチュートリアルとしては、実際の入力手順をアニメーションで見せるという明示的な方法を採用した。PC上のシステムで、「浜松市のファミレスに行きたい」などと自然言語で入力すると、画面上にあるカーナビシステム同様のメニュー表示の部分が、同じ条件で施設を検索する操作手順をアニメーションで示す。これで、ユーザはメニュー選択インタフェースで何をどのような順序で選択すればよいかを直接知ることができる。図5に自然言語入力インタフェースからメニュー選択型インタフェースへのチュートリアル画面の例を示す。

一方、カーナビでの、PCシステムへの自然言語入力のチュートリアルとしては、メニュー

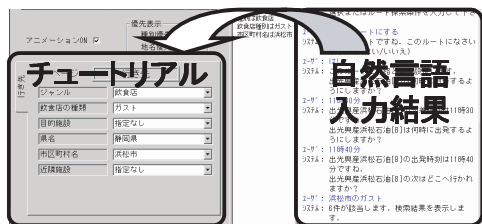


図 5 自然言語入力インタフェースからメニュー選択インタフェースへのチュートリアル
Fig. 5 Tutorial of the menu selection interface on the natural language interface.

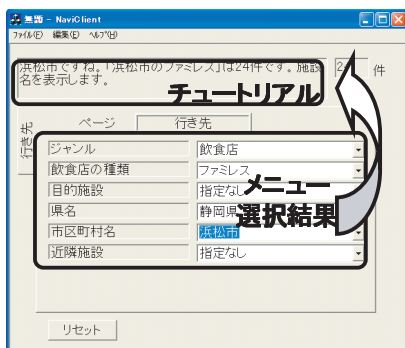


図 6 メニュー選択インタフェースからの自然言語インタフェースへのチュートリアル
Fig. 6 Tutorial of the natural language interface on the menu selection interface.

選択によって指定された施設検索条件を確認する応答文として、PC 上のシステムで同一条件の施設を効率良く検索できるような文を利用し、入力文に相当する部分を強調表示するという方法をとる。たとえば、ユーザがジャンルメニューから「飲食店」、飲食店の種類メニューから「ファミレス」、県名メニューから「静岡県」市区町村名メニューから「浜松市」を選択すると、システムは「浜松市ですね！浜松市のファミレス」は 件です」という応答をする。このようなやりとりを繰り返すことで、ユーザは目的の施設を効率良く検索するための自然言語表現を習得することができる。図 6 にカーナビシステムでの、自然言語入力インタフェースへのチュートリアルの例を示す。

ただし、施設検索機能において近隣施設からの距離は、PC 上のシステムでは「1.5 km 以内」など小数での指定が可能であるが、カーナビのメニュー選択では、「1 km 以内」「5 km

以内」など、整数値の 7 択となっており、小数値の指定はできない。このようにカーナビでは実行不可能な戦略が PC 上でとられた場合には、カーナビでは実行不可能であることを教えるために、操作デモを行わないことでメニュー操作での入力が不可能であることを示唆している。

5. DP システムにおける相互チュートリアルの評価

本章では、我々がこれまでに実現した、相互チュートリアルの効果に関する評価について述べる。

5.1 カーナビシステムから PC システムへのチュートリアル効果

まず初めに、応答文によるチュートリアル機能のないカーナビシステムを比較対象として、チュートリアル機能を搭載したカーナビシステムの PC システムへのチュートリアル効果を調べた。

5.1.1 実験方法

被験者は工学部系学生 20 名とした。被験者をチュートリアル機能のないカーナビシステムを使用してから PC 上のシステムを使用するグループ NT と応答にチュートリアル効果を持たせたカーナビシステムを使用してから PC 上のシステムを使用するグループ T とに分けた。2 つのカーナビシステムの差は応答文に PC システムへのチュートリアルが含まれているか否かのみとなっている。

被験者をそれぞれのカーナビシステムに十分に慣れさせるために、1 日 30 分を 3 日間、システム操作の練習を行わせた。その後、それぞれ使用したカーナビシステムに対する主観評価を行った後、すべての被験者に、あらかじめ用意したある家族の会話文（それぞれが旅行にあたっての希望などを述べている）を提示し、家族の希望を満たすドライブのプランを同じ PC 上のシステムで作成させた。なお、検索を途中で中止するためには「検索中止」と入力するなど、システム特有の操作や、どのような条件で検索が可能であるかといった説明はあらかじめ与えておいた。

5.1.2 実験結果

評価はプラン作成後に行った 7 段階評価のアンケートによる主観評価と、受理率やタスク達成時間などの客観評価で行った。まず、2 種のカーナビシステムに、それら単体では性能上の差がないことを確かめるためのアンケート調査を、各システムを使用したグループに対して行った。

この結果および、2 者間の t 検定結果を表 1 に示す。評価結果として、設問 1（使いやす

表 1 2種のカーナビシステムへのアンケート結果
Table 1 Questionnaire results for the two car navigation systems.

アンケート内容	NT	T	p 値
1. システムの使いやすさ	4.9	5.3	0.419
2. 応答の親切さ	4.9	5.2	0.624
3. 入力の際に迷ったか	3.6	3.4	0.633
4. 画面表示の分かりやすさ	5.0	5.6	0.382
5. 応答文を不快に感じたか	—	4.5	—

表 2 PCシステムに対するアンケート結果
Table 2 Questionnaire results for the PC system.

アンケート内容	NT	T	p 値
1. システムの使いやすさ	3.9	5.4	0.044
2. 応答の親切さ	4.1	5.3	0.046
3. 入力の際に迷ったか	2.3	3.8	0.002
4. 画面表示の分かりやすさ	5.0	5.5	0.486
5. カーナビの応答が役立ったか	—	6.2	—

い=7)ではNTの平均が4.9であるのに対してTが5.3,設問2(応答が親切=7)では4.9と5.2,設問4(表示が分かりやすい=7)では5.0と5.6と,Tの方がやや高い評価となっているが,t検定によると有意な差とはいえない程度である。また,設問3(入力の際に迷わなかった=7)では評価はほぼ同等であった。これにより,システム単体で使用する際にはユーザに与える印象にシステム間の差がないことが示された。

表2に,各カーナビシステムを使用してからPC上のシステムでプラン作成後に行ったアンケートの結果を示す。設問1,2,3でNTとTの間で有意差が見られた。NTとTそれぞれの平均は設問1で3.9対5.4,設問2で4.1対5.3と,チュートリアル機能のあるカーナビシステムを使用したTのほうが高い評価を得ることができた。これは,カーナビシステムからのチュートリアル効果によって,施設を検索する際にどのような文を入力すればよいか明確になり,システム全体としての使いやすさも向上したためと考えられる。また,設問3(入力の際に迷わなかった=7)でも同じく2.3対3.8と,Tのほうが,PCシステムに対して高い評価をしており,Tに対するチュートリアルの効果がうかがえる。また,Tのみに対して行った,設問5(応答が役に立った=7)」という質問では,平均6.2と高い評価を得ており,被験者もまたチュートリアルの効果を実感できていることが分かる。

表3に客観評価の結果を示す。ここでは全評価項目で有意差が見られた。PCシステムは施設検索機能以外にも経路探索機能や目的地の訪問予定時刻の設定機能などを備えている

表 3 PCシステムにおける客観評価結果
Table 3 Objective evaluation results for the PC system.

評価項目	NT	T	p 値
1. プラン作成時間(分:秒)	13:23	10:09	0.016
2. 入力回数(回)	24.3	11.2	0.005
3. キーワード数(個/文)	1.8	2.6	0.002
4. 入力文の受理率(%)	72.4	89.7	0.005

が,カーナビシステムは施設検索機能のみを備えているため,表3では,ドライブプラン作成全体にかかった時間である「1.プラン作成時間」以外は,施設検索に関する入力のための評価結果を記載している。

プラン作成時間はNTが13分23秒であったのに対してTが10分09秒と,NTよりもTのほうが約24%短くなった。また,項目2は施設検索に関する入力を行った回数の平均であり,NTが24.3回,Tが11.2回と,TはNTの半数以下となった。これは,項目3の1文あたりに含まれるキーワード数の平均からも分かるように,TのほうがNTよりも1度に必要な情報をより多く入力することで,効率良く施設を検索し,入力回数を減らすことができたためである。なおここでは,施設の種別や地域など,施設を検索するための検索条件を示す語をキーワードとしてカウントしている。

項目4は,ユーザの入力意図をシステムが正しく解釈できた割合である。結果は,NTが72.4%であるのに対してTは89.7%と,約17%向上した。この理由として,施設を検索するための条件として現在のシステムが対応していないものがNTでは多く使用されたが,Tではほとんど使用されなかったことがあげられる。対応していない検索の条件とは,「安いホテル」などデータベースに情報がない条件や「食事する」など目的のみの条件などである。この差はカーナビシステムからのチュートリアル効果によって,使用可能な検索条件が明らかになったためと考えられる。

これらの結果から,PCシステムで効率良く自然言語操作をするための入力文をカーナビの応答文に含めるというチュートリアルによって,自然言語入力を用いたシステムにおけるユーザが何を入力すればよいか分からなくて戸惑うという問題を解決できたと考えられる。

5.2 PCシステムからカーナビシステムへのチュートリアル効果

次に,PCシステムでのアニメーションの効果によるカーナビシステムへのチュートリアルの効果を調べた。

表 4 2種のPCシステムに対するアンケート結果
Table 4 Questionnaire results for two PC systems.

アンケート内容	NT	T	p 値
1. システムの使いやすさ	5.4	4.9	0.358
2. 応答の親切さ	4.5	5.1	0.399
3. 入力の際に迷ったか	4.6	4.1	0.221
4. 画面表示の分かりやすさ	4.8	5.3	0.540
5. アニメーションを不快に感じたか	—	3.9	—

5.2.1 実験方法

被験者は工学部系学生 20 名とした。前節で述べた実験と同様に、チュートリアル機能を持たない PC 上のシステムを使用してからカーナビシステムを使用するグループ NT とチュートリアル機能を備えた PC システムを使用してからカーナビシステムを使用するグループ T とに分けた。

それぞれのグループにはあらかじめ 1 日 30 分、3 日間 PC 上のシステムを使用させ、アンケートに回答させた。その後、すべての被験者にカーナビシステムのマニュアルを読ませたうえで、同じカーナビシステムを使用して、「現在の近くの病院を探す」など、6 種類の施設検索タスクを実行させ、アンケートによる主観評価と施設検索時間による客観評価を行った。

5.2.2 実験結果

本評価実験でも、施設検索後に行った 7 段階評価のアンケートによる主観評価と、タスク達成時間による客観評価を行った。初めに、2 種の PC システムに、それら単体では性能上の差がないことを確かめるためのアンケートを行った。この結果及び、2 者間の t 検定の結果を表 4 に示す。T に対して行った、アニメーション表示を不快に感じたか（不快でなかった=7）という質問に対しては平均 3.9 と中間値よりもやや悪い程度の評価であった。しかしその他の質問項目に関しては、ほぼ同等であり、チュートリアル機能のためのアニメーション表示が PC システムに対する主観評価に対して影響を与えていないことが示された。

表 5 に、各 PC システムを使用してからカーナビシステムで施設を検索した後に行ったアンケートの結果を示す。2 つのグループ間でアンケート結果に大きな差は現れなかった。しかし、T に対する、PC システムのアニメーション表示がカーナビの操作に役立ったかという質問への回答は平均 5.9 と、被験者がチュートリアルの効果を実感していることがうかがえる。

表 6 には各タスクの実行に要した時間を示す。タスク 1 は単純なタスクであったため、大

表 5 カーナビシステムに対するアンケート結果
Table 5 Questionnaire results for the car navigation system.

アンケート内容	NT	T	p 値
1. システムの使いやすさ	4.7	5.1	0.587
2. 応答の親切さ	5.0	4.6	0.529
3. 入力の際に迷ったか	3.3	2.9	0.540
4. 画面表示の分かりやすさ	5.1	5.1	—
5. アニメーション表示が役立ったか	—	5.9	—

表 6 カーナビシステムでの施設検索所要時間
Table 6 Time required for facility search on the car navigation system.

タスク	NT (分:秒)	T (分:秒)	p 値
1. 浜松市のガスト	0:47	0:37	0.529
2. 現在の近くの病院	0:40	0:30	0.142
3. 浜名湖バルバルの近くのホテル	2:08	1:03	0.028
4. 駅の近くの公園	1:10	1:12	0.905
5. 現在地から 30 分以内のローソン	1:14	0:47	0.026
6. 浜松城から 10 km 以内の警察署	2:13	1:35	0.256

きな差は見られない。タスク 2 では基準地の指定が必要となり、やや複雑さが上がり、両グループでの差が広がる傾向が見られる。タスク 3 は、基準地の指定に複数操作が必要で、タスク 1, 2 と比べ約 2 倍の操作回数を要する。このタスクにおいては、両グループ間で優位な差が見受けられる。タスク 4 は、タスク 2 とほぼ同一の操作で達成できるものであり、タスク 2 を経験した直後であったために、慣れの影響で両グループ間に差が出ていないものと思われる。なお、タスク 4 がタスク 2 の類似処理であり、慣れの影響があるにもかかわらず、タスク 4 の平均タスク達成時間がタスク 2 の 2 倍程度となっているのは、メニューから「病院」を指定するのにかかる時間と「公園」を指定するのにかかる時間に大きな差があったためである。「公園」を指定するためには、まず施設カテゴリから「レジャー施設」を選択しなければならないが、これに気づかず「公共機関」など他のカテゴリのメニューで公園を探し、見つけれずに後戻りをするケースが多くあった。一方、「病院」の場合、ユーザがカテゴリの判断に迷うことが少なかった。タスク 5 では、所要時間（あるいは距離）の指定を必要とし、複雑さがさらに増している。このタスクにおいても両グループ間で有意差が見られた。タスク 6 はタスク 5 よりも基準地の指定で複雑さが高く、有意な差とはいえないものの平均時間に差が見られた。

これらの結果から，チュートリアルの効果によって，特に複雑な操作の場合に，ユーザがメニュー入力の操作手順を覚える煩雑さを軽減することができたと考えられる．

6. ま と め

本論文では，複数の情報端末機器における異なるインタフェースを使用して同一システムの同一機能を利用する際に，あるインタフェースからの操作方法を習得していても他のインタフェースからの利用方法は不明なため，機器を変更するたびにそのインタフェース独自の操作方法を個別に習得しなければならないという問題に対して，ミラー効果に基づくインタフェース間の相互チュートリアル機能による解決を試みた．

相互チュートリアルを実現した DP システムにおいて，一方のインタフェースでシステムを使い慣れており，他方のインタフェースは未経験であるがデバイスの基本的利用方法は知っているユーザに対して行った評価実験により，未経験のインタフェースを初めて利用する際のタスク達成時間の短縮と入力受理率向上に対して効果があるといえる結果が得られた．同一の機能を複数のインタフェースから利用されるシステムで，同様のユーザを想定する場合，ミラー効果に基づく相互チュートリアルにより同様の効果が期待できる．

本論文で提案する相互チュートリアルでは，特定の機能を利用するために利用される可能性の高い戦略を効率的に学習させるという，同一システムを扱う利点を生かした枠組みを採用している．これによって，複数の端末からアクセス可能な統合システムを使用することの利点を感じられ，端末を変更しても同一のシステムが選択されやすくなると考えられる．

またユビキタス環境が整備されるにつれて，同一のシステムを，状況に応じて携帯電話や PDA などのモバイル端末や，デスクトップ PC のような固定端末など，様々な機器から利用する形態は今後も増えることが予想される．そういった環境では，複数のインタフェースをスムーズに使い分ける必要がさらに高くなる．その際にも，本論文で示したミラー効果による相互チュートリアルは有効であると考えられる．

参 考 文 献

- 1) NISSAN CARWINGS : 日産カーウィングス，日産自動車株式会社 (オンライン)．入手先 <http://drive.nissan-carwings.com/WEB/> (参照 2008-07-01)
- 2) 歓迎会・送別会・宴会・パーティ グルメ情報検索サイトぐるなび，株式会社ぐるなび (オンライン)．入手先 <http://www.gnavi.co.jp/> (参照 2008-07-01)
- 3) 桂川景子ほか：ドライブプラン作成・編集のための PC 版サブシステム DPS-PC の構成と評価，情報処理学会論文誌，Vol.44, No.12, pp.2990-3001 (2003)．

- 4) Katsuragawa, K., Niwa, M., Yanagi, T., Watanabe, M., Itoh, T., Konishi, T. and Itoh, Y.: Constructing an Interactive Natural Language Interface for a Drive Planning System, *SNLP COCODA*, pp.237-242 (2002)．
- 5) Nielsen, J.: *Usability Engineering*, Academic Press (1993)．篠原稔和 (監訳)，三好かおる (訳)：ユーザビリティエンジニアリング原論—ユーザの為のインタフェースデザイン，東京電機大学出版局 (2002)．
- 6) 松井弘行，横田公幸，徳永幸生：電話操作におけるメンタルモデル，電子情報通信学会論文誌 D，Vol.J70-D, No.11, pp.2058-2064 (1987)．
- 7) ISO9241-11: 1998 Ergonomic requirements for office work with visual display terminals (VDTs) – Part 11: Guidance on usability, ISO (1998)．
- 8) Shneiderman, B. : ユーザインタフェースの設計：使いやすい対話型システムへの指針，日経 BP 社 (1987)．
- 9) Foley, J.D., et al.: *Computer Graphics – Principles and Practice*, Addison-Wesley (1990)．佐藤義雄 (訳)：コンピュータグラフィックス理論と実践，オーム社 (2001)．

(平成 20 年 3 月 31 日受付)

(平成 20 年 10 月 7 日採録)



桂川 景子 (正会員)

2003 年静岡大学大学院情報学研究所情報学専攻修了．同年日産自動車 (株) 入社．2006 年静岡大学創造科学技術大学院自然科学系教育部情報科学専攻入学．音声言語インタフェースの研究開発に従事．



大野 健 (正会員)

1989 年筑波大学第三学郡基礎工学類卒業．1991 年同大学大学院理工学研究科物理工学専攻修士課程修了．同年日産自動車 (株) 入社．車両用距離センサや音声インタフェースの研究開発に従事．



富樫 実 (正会員)

1980年東京工業大学大学院情報工学専攻修士課程修了。同年日産自動車(株)入社。車両用光通信システム,故障診断システム等の開発に従事した後,音声対話インタフェースの研究開発に従事。電子情報通信学会,人工知能学会各会員。



小暮 悟 (正会員)

2002年豊橋技術科学大学大学院工学研究科電子・情報工学専攻博士後期課程修了。同年豊橋技術科学大学工学部研究員。同年愛知教育大学教育学部情報教育講座助手。2004年静岡大学情報学部情報科学科助手。現在,同大学同学科助教。音声対話システム,対話管理,対話制御に興味を持つ。博士(工学)。



伊藤 敏彦 (正会員)

1999年豊橋技術科学大学大学院工学研究科博士後期課程電子・情報工学専攻修了。同年静岡大学情報学部情報科学科助手。2004年北海道大学情報科学研究科メディアネットワーク専攻助教授。現在,同大学同学科准教授。音声対話システム,対話制御,車載情報システム等に興味を持つ。博士(工学)。電子情報通信学会,人工知能学会,日本音響学会各会員。



小西 達裕 (正会員)

1987年早稲田大学理工学部電子通信学科卒業。1992年同大学大学院博士後期課程修了。1991年早稲田大学理工学部情報科学科助手。1992年静岡大学工学部情報知識工学科助手。現在,同大学情報学部情報科学科准教授。知的教育システム,知的対話システム等に興味を持つ。博士(工学)。電子情報通信学会,教育システム情報学会,日本認知科学学会各会員。



伊東 幸宏 (正会員)

1980年早稲田大学理工学部電子通信学科卒業。1987年同大学大学院博士後期課程修了。同年同大学理工学部電子通信学科助手。1990年静岡大学工学部情報知識工学科助教授。2000年同大学情報学部教授。現在,同大学創造科学技術大学院教授(情報学部兼務)。工学博士。自然言語処理,知的教育システム等に興味を持つ。電子情報通信学会,言語処理学会,教育システム情報学会,日本認知科学学会各会員。