

確率的な提示に基づくインタラクションの誘発手法

水口 充

情報通信研究機構

常に何らかの情報を提供し続けてユーザを日常的に支援するシステムにおいて、複数の情報を属性に応じた確率で自動的に切り替えて提示することによってユーザに情報の存在を気付かせ利用を促す手法を提案する。この手法はユーザが提示された情報を見逃すことに配慮し、情報の存在を確実に伝えるのではなく有用な情報を利用できる可能性を向上させることを目標としている。本稿では本手法について基本的な考察を行い、具体的な実装例について説明する。

Interaction Inspiration by Probabilistic Presentation

Mitsuru Minakuchi

National Institute of Information and Communications Technology

We propose a probabilistic presentation method of information for everyday computing systems which provide various information for users continuously. This method chooses presented information according to probabilities which reflect characteristics of information, and inspires users to interact with the system by reminding presented information. Considering that users might miss presented information, we set the goal of the method to improving the possibility of utilization of useful information, not making users aware of information certainly. In this paper, we discuss bases of the method and explain an implemented system.

1 はじめに

ユビキタスコンピューティングやウェアラブルコンピュータ、情報家電などが実用化されつつあり、大量のデジタルコンテンツやネットワークサービスを日常的に利用できるようになってきた。

一方、現在のユーザインタフェースの主流である直接操作による対話型システムにおいては、ユーザは何らかの処理結果を得るためには能動的に操作を行う必要がある。しかし一般的な日常生活においてユーザは、何らかの作業を行って手を離せない、リラックスしていて積極的に操作することを煩雑に

感じる、特に要求を明確にしていない、などの様々な状況にあることが多い。このため従来のパーソナルコンピュータのユーザインタフェースは特定の作業を行うことには適しているものの、日常生活を支援する目的には不都合があると言える。

コンテキストウェアコンピューティングは、システムがユーザの状況（コンテキスト）に適した情報を推測して提供することによって、ユーザを支援することを目指すものである。しかしユーザの潜在的な欲求を正確に推測することは難しい。また推測が正確であっても、ユーザはシステムをお仕着せがましく感じかねないという問題もある。

この問題に対し、我々は受動型インタラクションスタイル [4] を提案している。この基本コンセプトは、ユーザとシステムとのインタラクションが疎な状態においてシステム側からの働きかけをきっかけとしてインタラクションが密な状態への移行を促すことにある。すなわち、ユーザへの気付かせによるインタラクションの誘発を特徴としており、情報の取捨選択や操作の主導権をユーザに残すことによって上述の問題の解決を目指している。

ところで、受動型インタラクションスタイルによる日常生活を支援するシステムにおいては、ユーザに情報の存在を確実に伝えるのではなく、有用な情報を利用できる可能性を高めたり普段は気付かないような情報を伝えることに主眼をおくのがよいと考えられる。なぜなら、ユーザはシステムが提示する情報を見逃しているかもしれないが、システムはユーザの活動を強要することはできないからである。更には、確実に情報を利用できるようにするためには強制的な通知を行ったり、従来の対話型システムを利用すればよいからである。

しかしながら、ユーザにとって有用な情報ほど利用できる可能性が高い方が望ましい。そこで本稿では、複数の情報を属性に応じた確率で提示することによって利用を促す手法を提案する。また、日常的に利用するシステムのためのユーザインタフェース環境として実装を行っているので報告する。

2 確率的な情報提示

2.1 想定するシステムと提示方法

本稿では日常的に利用するシステムとして、周辺ディスプレイ (peripheral display) と称されるような提示方法を用いてユーザのインタラクションの有無に関わらず常に何らかのサービス¹をユーザに提供し続けるシステムを想定する。具体的な装置としては、

¹本稿ではアプリケーション、コンテンツ、メッセージングなどのコミュニケーション、インターネットサービスなど、ユーザに対して何らかの価値を与えるものを総称してサービス、サービスによって扱われる情報をサービスデータと呼ぶ。

壁掛け型のディスプレイや壁面に投影するプロジェクタ、あるいは置物のように部屋の片隅に置かれているコンピュータなどである。

このようなシステムによって提供されるサービスは、ワープロや表計算などといったユーザが能動的に作業を行うためのアプリケーションよりも、様々な情報を提供するアクセサリ的なサービスが主となるべきである。このようなサービスとしては例えば次のようなものが挙げられる：

- デスクトップアクセサリ：時計、カレンダーなど
- 情報配信：ニュース、天気予報、株価など
- 行動の管理、支援：スケジュール管理など
- コミュニケーション：電子メール、インスタントメッセージなど
- コンテンツの視聴：音楽プレイヤー、アルバムのスライドショーなど

これらのアクセサリ的なサービスを従来のパーソナルコンピュータ上で利用する際には、メインの作業を中断しないように、タスクバーのように常に前面に表示される領域を占有して常時提示する方法や、ポップアップウィンドウなどを通じて随時提示する方法 [5] などが用いられている。また、参照したい部分を他のウィンドウで隠さないようにユーザが手作業で配置して利用することも多い。更に、通知音やダイアログウィンドウを通じてユーザに処理状況の変化や新着の情報の存在を伝えている。

しかし、日常的に利用するシステムにおいては、ほとんどの時間はユーザとシステムとの間のインタラクションが疎な状態、すなわち、ユーザはシステムにほとんど注意を払っていない状態にあると予想される。従って、上述のようなタスク切り替え操作をユーザが行うことは期待できないし、小さな領域に表示された複数の情報から必要な情報を認識することは難しい。

そこで、次のような方針でユーザにサービスを提示することを想定する。例えて言うなら、ニュース記事を流し続ける電光掲示板のようなものである。

- 常に1つのサービスデータを画面に表示し、同時には複数のサービスデータは表示しない。これはユーザの認知負荷を軽減し、画面を一瞥するだけでサービスの内容を理解できるようにするためである。
- 表示するサービスデータは所定の時間ごとに別のサービスデータに切り替える。
- ユーザは操作によってテレビのチャンネルを変えるように、表示されているサービスデータを別のサービスデータに切り替えることができる。
- 必ずユーザに提示しなければならないサービスデータは、上記の自動的な切り替えやユーザの操作による切り替えに優先して表示する。
- システムは必要に応じて音を利用する。表示されているサービスデータが切り替わる時や、特に重要あるいは緊急な情報を表示する際にユーザの注意をひく時などである。

ところで、このような常時サービスを提示し続ける方法でユーザにとって意義のあるサービスデータをすべて扱うことが可能かどうかを検討してみる。まず、一般的なユーザー一人が利用するサービスデータをおおよそ次のように見積もってみる：

- スケジュール：数十～数百個
- メール／メッセージ：数十～数百個
- ウェブページ（ニュースサイトの記事は下記のニュースに含める）：～数十個
- ニュース：数十～数百個
- 進行中の作業に関するファイル、アプリケーション：～数十個
- オンラインサービス：～数十個
- 実世界に関係するサービス²：～数十個
- 写真や音楽のプレイヤー：上限無し

²例えばドアホンのモニターやエアコンの温度調節などの、実世界指向あるいはユビキタスコンピューティング的なアプリケーションサービス。

写真や音楽の場合は蓄積あるいはストリーミング配信されるコンテンツすべてが提示して意義のあるサービスデータとなりうるので上限はない。しかし、ユーザにとって特別な意義を持つコンテンツはそれほど多くはなく、また、どのコンテンツであっても視聴を想起する可能性はあると考えられる。よって、それぞれアルバムビューアと音楽プレイヤーというサービスにまとめて扱うことにする。同様にウェブページもコンテンツ自体は無限に存在するが、よく閲覧するページはそれほど多くないであろう。

これらを合計して、一人のユーザが利用するサービスデータをおおよそ数百～千程度と見積もる。もちろんこれらの数字には個人差があるが、爆発的に増加することはあまり無いであろう。

一方、ユーザが画面を見ることが可能な時間は、ユーザの周りにある様々なデバイスを駆使できるユビキタス環境を前提とすれば、最大限で24時間から睡眠時間などを除いた時間となる。しかし現実的には、ユーザの意識が何かの作業などに集中している間は周辺ディスプレイに表示されている内容に気付かない可能性が高い。そこで控えめに見積もって、仕事中の8～10時間のうちの1時間程度と、仕事以外の6～8時間のうちの3時間程度の、計4時間程度は提示されたサービスデータに気付くことが可能な時間とする。この時間も個人の活動パターンに大きく依存するが、最低ラインとしては十分可能な値であろう。

約4時間の間に約千個のサービスデータすべてを均等な時間だけ提示すると、一つあたりの提示時間は約14秒となる。これはテレビコマーシャル一つの平均的な放映時間に近く、サービスデータの内容をユーザに伝えることは十分可能である。

現実的にはあるサービスデータを一日に一回十数秒だけ提示してユーザに気付かせることは困難である。しかしこの見積もりでの提示時間は最低ラインである。また、サービスデータの多くは見逃しても支障がないような内容であろうし、必ず気付く必要があるサービスデータについては強力な通知を行えばよい。

以上の検討より、常時サービスを提示する方法によって、個人にとって意義のあるサービスデータのほとんどを扱うことは可能であると言える。

2.2 提示確率の算出方法

前節で検討したように、ユーザにとって意義のあるサービスデータは数多く存在するが、その意義はまちまちである。また、ユーザが画面を見るタイミングはランダムで予測できないとすると、あるサービスデータの総提示時間は長いほどユーザが見る可能性が高くなる。そこで有意義なサービスデータほど多くの提示時間を割り当てて気づきやすくすることを考える。

総提示時間は1回あたりの提示時間と提示回数の積となるが、ここでは1回の提示時間は固定とする。なぜなら、連続して長時間提示してもユーザの状況が変化しなければ気づかない可能性が高いし、提示内容にアニメーションや動画を利用する場合に提示時間を可変にするのは困難があるからである。

すると、サービスデータの有意義さが高いほど提示回数が多くなるようにすればよいことになる。しかしサービスデータの意義はユーザのコンテキストに依存する。そこで、システムが提示しているサービスデータを切り替える時点において、ユーザにとってどれだけ意義があるかを表すパラメータである提示スコアをそれぞれのサービスデータごとに算出し、次に提示するサービスデータを提示スコアに比例した確率で選択することにする。このようにすればコンテキストに配慮しつつ、ユーザにとってお仕着せがましくない情報提示が可能となる。

サービスデータの意義は重要度と緊急度とで表現できると考え、提示スコアは重要度と緊急度の積であるとする。

重要度はサービスデータの必要性やユーザに与える影響の大きさを表現するパラメータである。例えば、必ず出席しなければならない会議の予定の重要度は高いし、新製品の紹介のメールの重要度は低いであろう。また、重要度はコンテキストに依存して

変動することもある。例えば、電車の時刻表データの重要度はユーザが駅にいるときに高くなる。なお、時間に関するコンテキストは次に説明する緊急度で扱うことにする。

サービスデータの重要度を評価するためには、データ自体に明示的に付与された値や、ヒューリスティックな知識や内容解析、あるいはベイジアンフィルタやSVM (Support Vector Machine) などのツールを用いることができる。また、コンテキストに応じて変動する重要度の評価方法を定義する必要がある。

緊急度はユーザがサービスデータを見るべきタイミングを表現するパラメータである。緊急度はサービスデータに設定された時刻と現在の時刻との関係で評価されるが、その評価方法はサービスに依存する。例えば、スケジュールデータの場合は開始時刻が近づくにつれて緊急度が増大するし、写真や音楽のプレイヤーの緊急度は変化しないと考えられる。また、緊急度の変化の仕方もサービスデータによって異なっており、例えばニュースのような速報性の高いサービスデータほど急激な変化となる。

重要度と緊急度のいずれも客観的に正しく評価することは難しいが、ここでは理想的に評価することが可能であると仮定する。現実的には提示は確率的に行われるので、評価は十分に妥当であれば必ずしも完全である必要はない。

ところで、ユーザが表示されているサービスデータに気付いたときにはユーザの注意は表示画面に向けられているので、引き続き表示されるサービスデータも見ている可能性が高くなると予想される。この時、例えば東京出張の予定の次に東京の天気予報が提示されるというように、連続して提示されるサービスデータ間に関連性があるとユーザにとって理解しやすい。そこで、直前に提示されていたサービスデータとの関連度を導入して、提示スコアに加味することにする。なお、関連度についても客観的に正しく評価する方法を定義することは難しいが、例えばサービスデータに付与あるいは抽出されるキーワード群間の類似度などの方法で理想的に算出できるものとする。

以上をまとめると、ある時点での i 番目のサービスデータの提示スコア S_i を、重要度 I_i 、緊急度 U_i 、関連度 R_i とから、 $S_i = k_1 I_i U_i + k_2 R_i$ として計算し (k_1, k_2 はそれぞれ係数)、提示される確率は $S_i / \sum S_i$ とする。

2.3 割り込み

上述のような確率的な提示方法は、サービスデータの提示のタイミングを確実にコントロールするものではない。すなわち、重要度が高いからといってユーザーが確実にサービスデータを見ることは保証できないし、緊急度が高いからといって即座にサービスデータが提示されることは保証されない。想定しているシステムの使われ方においてはインタラクションのきっかけを与えれば良いので確実な提示を求める必要は無いが、時報のように特定の状況でしか意味の無いサービスデータも扱える方が望ましい。

そこで割り込みの機構を用意することにする。具体的には、サービスデータが割り込み条件を保持できるようにし、システムは条件が満たされた場合には提示するサービスデータを強制的に切り替えるようにする。

また、インタラクションが密な状態に移行したとき、すなわちユーザーが提示されたサービスデータによって積極的な利用を誘発された際には、サービスデータの自動提示を中断してインタラクティブな操作を可能にする。ユーザーによる操作としては、提示されているサービスデータを明示的に切り替えるための操作と、サービスを利用するための操作が想定される。いずれの操作が発生した場合も自動的なサービスの提示は中断し、操作が一定時間無かったら再開するようにすればよい。

3 実装

前章で説明した方法の実現可能性と有用性を検証するために、我々は日常的に利用するシステムのためのサービスプラットフォーム i2SPEC: Interaction

Inspiring Service Platform for Everyday Computing を、Java(JDK1.4) を使用して構築している。

3.1 構成

図1は i2SPEC の概念的な構成図を示している。以下、各構成について説明する。

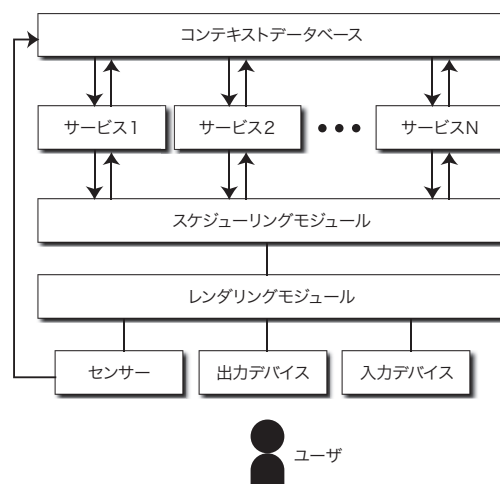


図1: 概念構成図

サービスモジュール サービスモジュールは従来のアプリケーションやネットワークサービスのような、サービスの処理を行うものである。サービスモジュールの役割は次の通りである：

- 具体的なサービスの処理を行う。
- サービスデータのクラスインスタンスを生成してスケジューリングモジュールに登録する。
- サービスデータの重要度と緊急度の算出方法および割り込み条件を定義する。
- サービスデータの提示方法（表示、音、提示継続時間など）を決定し、サービスモジュール内で、あるいはレンダリングモジュールを通じて表示／再生する。
- ユーザーが積極的に利用する際のインタラクション手段を提供する。

スケジューリングモジュール スケジューリングモジュールは2章で述べたような方法で、サービスモジュールによって登録されたサービスデータの中から次に提示するものを選択する。

レンダリングモジュール サービスデータを提示するための画面表示や音の再生機能を提供する。

現バージョンでは、SVG(Scalable Vector Graphics)[6] 文書を表示する機能を Batik SVG Toolkit[1] を利用して実装している。SVG は XML(eXtensible Markup Language) に準拠しているので、プログラム内から DOM (Document Object Model) API を通じてグラフィクスデータの内容を変更することができる。これを利用して、加工の対象となる要素に特定の ID を付与したテンプレートの SVG 文書を用意しておき、サービスデータの内容に応じて加工して表示することができる。この機能により各サービスモジュールは個別の描画処理を実装しなくてもグラフィクスエディタで別途作成した表現力の高い多様な表示画面を扱うことができる。また、エンドユーザが表示画面をカスタマイズすることも容易となる。

コンテキストデータベース ユーザや外界の状況、およびサービスの履歴や状態などの、コンテキストデータを管理する。コンテキストデータは各種センサーによって獲得される他、サービスモジュールやスケジューリングモジュールの動作状況やユーザの閲覧・操作履歴を扱うこともできる。コンテキストデータベースに保持されているコンテキストデータはスケジューリングモジュールがサービスデータの割り込み条件を解決するため、および、サービスモジュールがサービスデータの重要度および緊急度を算出するために参照される。

現段階では、様々なサービスを想定して汎用的にコンテキストデータを扱うことが難しく、実装が不十分である。この問題に対し、我々はコンテキストデータを記述するための枠組みについて検討中である。

3.2 動作例

i2SPEC の動作を検証するために、いくつかのサービスモジュールのサンプルを実装している。

時計 現在の時刻を表示する。サービスデータは時計サービス自体を表現するものとして、一つだけを扱うようにした。つまり、このサービスデータの内容は現在の時刻となる。

常に時刻を表示するという点では本物の時計の方が有用であり、i2SPEC 上で偶発的に時刻を表示することはあまり重要ではないと考えられるので、サービスデータの重要度は普段はあまり高くないように設定した。また、1時間ごとの定刻に近くなるほど緊急度が高くなるようにして、時報を高い確率で知らせることができるようにした。

スライドショー 特定のディレクトリに格納されている画像データを順次切り替えて表示する。スライドショーについても扱うサービスデータは一つだけとし、このサービスデータが提示の対象となったときに表示する画像を選択するようにした。この方法では画像の数が増えてもスケジューリングモジュールに負荷をかけないという利点がある一方で、ある時点で提示されているサービスデータと個別の画像との関連度を扱うできないという欠点がある。

現在の実装ではサービスデータの重要度と緊急度は共に低く設定したが、将来的にはコンテキストに応じて、例えばユーザがリラックスしているときには重要度を上げて表示される確率を高くすることが考えられる。

メールチェック メールボックスをチェックしてメールの概要を通知する。IMAP 形式のメールサーバに対応しており、それぞれのメールごとにサービスデータを生成する。便宜的に、重要度は新着、差出人や宛先のアドレス、スパムフラグの各情報から、緊急度は送信時刻から算出するようにした。実用的にはメールの内容解析を利用するのがよいであろう。

スケジュール通知 ネットワークスケジュールサーバ上に管理されているスケジュールを通知する。各スケジュールデータをサービスデータとして扱う。重要度はスケジュール自体に設定されている重要度を参照し、緊急度はスケジュールの開始時刻と終了時刻とから算出するようにした。

RSS リーダ WWW 上のニュースやウェブログなどの記事を取得して表示する。記事ごとに個別のサービスデータとして扱う。現在の実装では重要度と緊急度は固定としている。将来はサービスデータごとに重要度を記事の内容と予め登録されたユーザの興味との一致度に応じるように設定し、緊急度を記事が生成された日時からの経過時間が長くなるほど低くなるように設定する予定である。

図 2 および 3 は、実際にサービスを提示している画面の例である。これらの画面が適宜切り替わって継続的に提示される。



図 2: 時計の表示例



図 3: スケジュールの表示例

4 議論

現在 i2SPEC は構築の途上にあるが、試験的な運用を通じて様々な課題が明らかになっている。

複数のユーザの配慮 居間などの共有スペースで運用するにはユーザが複数存在する場合に配慮することが重要である。例えばメールやスケジュールの通知サービスの場合、サービスデータが誰に関係しているのか分かりにくいし、人に見られると困るという場合もある。この問題は、複数のユーザが同室に存在する場合はプライベートなサービスデータを提示の対象としない、あるいは提示する際には詳細な内容を省く、というようにコンテキストを利用することで解決できる。どのユーザがシステムを参照可能な範囲にいるかを検出する仕組みは現在の実装していないが、ユーザの検出センサや明示的なログイン操作を利用すればよいと考えている。

提示スコアの一貫性 i2SPEC ではサービスモジュールは個別に開発が可能であり、サービスデータの重要度や緊急度の算出方法はサービスモジュールごとに決定するようにしている。このためサービス間で提示スコアの一貫性が保たれないという虞れがある。例えば最重要な予定とメールとではどちらがより重要かを決めることは難しい。しかし前述のように本手法はサービスデータを確実に提示することを目指すものではないので、できるだけ一貫性がある方がよいものの、完全である必要はないと考えている。現在の i2SPEC では、重要度についてはガイドラインを定め、緊急度については幾つかの算出関数を提供³することによって対応している。

密なインタラクションへの移行 現在は、提示されたサービスデータを密に利用する仕組みは実装していない。これを実現するためには、例えばメールが提示されているときにボタンを押すと、メールアプリに切り替わって詳細を読んだり返信を作成できる

³現在の実装では定数型、比例型、指数型を用意している。

というように、対話型のユーザインタフェースに切り替えればよいと考えている。

5 関連研究

Horvits は予測の確度に応じてエージェントの振る舞いを変えることによる自動実行と直接操作の融合を提案している [2]。インタラクションが疎な状態と密な状態とで挙動を切り替える点や、確度に配慮している点で本研究のアプローチと共通している。一方で、本研究はサービスデータをユーザに気付かせることを基本としている点で異なっている。

周辺ディスプレイを介した情報提示方法は多数提案されている。例えば、MacIntyre らは、作業に関連する文書の集まりをアイコン化して壁面に投影し、作業の存在を気付かせるシステムを提案している [3]。しかし、情報の重要度に応じてユーザが気付く可能性を確率的に制御するアプローチは従来なかった。

Scope[7] は作業中のパソコンの画面の隅に表示される円形のウィンドウを通じて複数のサービスデータの存在を気付かせる試みである。メール、カレンダー、TO DO、その他の通知のサービスデータがそれぞれ、内容の特徴を反映したアイコンで表現され、緊急度に応じて配置される。しかし、アイコンによる表現はインタラクションを行うまで内容を確認できない。また、同時に扱うことのできるサービスデータには限りがある。一方 i2SPEC はインタラクションが行われない環境での利用を前提としているため、サービスデータの内容を提示するようにしている。

6 おわりに

本稿では情報の内容に応じて確率的に提示することによってユーザに情報の存在を気付かせ、情報とのインタラクションを誘発する手法について説明した。本手法は、ユーザが提示された情報を見逃すことを考慮している点を特徴としている。つまり、本手法による情報提示システムは必ずしもすべての情報を提示するとは限らないので、ユーザは強要され

ることなく気軽にシステムを利用することができる。これは日常的に利用するシステムにとって重要な要件であると考えている。

本手法の有効性については継続的な運用によって評価する必要がある。今後、サービスモジュールを拡充して実用性の高いシステムを構築していきたい。

謝辞

本研究の一部は情報処理推進機構平成 15 年度未踏ソフトウェア創造支援事業の補助によるものである。

参考文献

- [1] Batik SVG Toolkit, <http://xml.apache.org/batik/>
- [2] Horvits, E.: Principles of Mixed-Initiative User Interfaces, in *Proceedings of CHI'99*, pp. 159–166 (1999).
- [3] MacIntyre, B., Mynatt, E., Volda, S., Hansen, K., Tullio, J. and Corso, G.: Support For Multitasking and Background Awareness Using Interactive Peripheral Displays, in *Proceedings of UIST'01*, pp. 41–50 (2001).
- [4] 水口 充, 竹内 友則, 渋谷 雄, 倉本 到, 辻野 嘉宏: 受動型インタラクションスタイル: 密なインタラクションへの移行を促すユーザインタフェース, *ヒューマンインタフェース学会論文誌*, Vol. 5, No. 4, pp. 519–528 (2003).
- [5] Mypop, <http://www.mypop.jp/>
- [6] Scalable Vector Graphics (SVG) 1.1 Specification, <http://www.w3.org/TR/SVG/>
- [7] van Dantzich, M., Robbins, D., Horvitz, E. and Czerwinski, M.: Scope: Providing Awareness of Multiple Notifications at a Glance, in *Proceedings of Advanced Visual Interfaces 2002* (2002).