

ユーザ習熟度判定のための情報の記述と解析

7J-3

野見山 健* 長崎 等 東 基衛
早稲田大学理工学部

1 概要

従来、ユーザインタフェース、特にオンラインヘルプはその使用性に大きなバラつきがあった。その理由は多岐にわたるが、その中でも本研究ではユーザの習熟度に適したオンラインヘルプが提供できていないことに着目し、ユーザ習熟度を判定するために必要な情報は何かを明確にし、その記述と解析方法を示す。

また、本研究で示す方法によりどのようなアプリケーションについて解析可能かどうかの適用範囲についても考察する。

2 背景及び従来研究

ユーザの特定アプリケーションへの習熟度合をユーザ習熟度と言う。アプリケーション各々についてのユーザ習熟度を計測できれば各々のユーザに適したオンラインヘルプの提供が可能である。しかし、ユーザ習熟度をどのようにしたら計測可能なのかについては定式化されたモデルがなく、どのようなアプリケーションなら計測可能なのかなどについてもよく分かっていなかった。

ユーザ習熟度の概念と定義は諸説があるが、その中でも本研究では長崎他による[NAGA93]で定義されたユーザ習熟度の考え方を採用する。この方法ではユーザの行動からユーザ習熟度の測定を行う。この方法の優れている点はユーザの行動をユーザの目的である「アクション」と実際の操作である「メソッド」に分けた点である。これによってユーザ習熟度を的確に判定ができる。「メソッド」は1つないし複数の「ユーザ操作要素」から構成される。「ユーザ操作要素」とはアプリケーションを使用する際にユーザが行える最小単位の基本操作(メニュー項目の選択、対象物の選択など)である。

3 問題点

[NAGA93]ではユーザ操作要素が取り出すことができれば、ユーザ習熟度が計測できる、というしくみになっていたが、ユーザ操作要素をどのようにして取り出すかについては言及されていなかった。実際の問題としてユーザ操作要素を取り出す方法と、ユーザ操作要素からメソッドを抽出する方法についてはOSやGUIカーネルからの情報をそのまま用いただけでは難しい。例えば、文章のセンタリングをとっても、センタリングの機能を選択する、スペースを挿入する、タブを挿入するといった複数のメソッドがあり、特に後者の場合、OS, GUIカーネルからの直接の情報では普通の文字入力と認識されてしまうので、この情報をそのままメソッドとみなすことはできない。

4 モデル

本研究で提案するモデルは図1のようなものである。

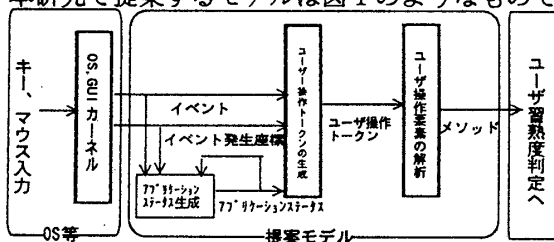


図1 提案モデルとOS等との関係

キーボード、マウスからの入力はOS, GUI等のカーネル部分を通して、「イベント」に変換される。イベントはその属性として様々な情報を含むのが普通だが、そのうち特に重要なのはイベントの発生場所についての情報である「イベント発生座標」である。

イベントとイベント発生座標、を基にして「アプリケーションステータス」の生成が行われる。

アプリケーションステータスとはアプリケーション内部が持っている現在の状況(メニュー選択中である、範囲指定であるなど)である。アプリケーションステータスを生成する際にはイベントとイベント発生座標以外にも直前のアプリケーションステータスが何なのかが必要である。

*A Study of Description and Analysis for Rating User's Skill Level
Takeshi Nomiya, Hitoshi Nagasaki, Motoei Azuma
Waseda University
3-4-1, Okubo, Shinjuku-ku, Tokyo, 169, Japan

ユーザ操作トークンの生成ではイベント、イベント発生座標、アプリケーションステータスの情報を基にユーザ操作要素を加工したユーザ操作トークンを作る。具体的にはイベント、イベント発生座標、アプリケーションステータスの直積とする。

生成したユーザ操作トークンを解析することでユーザのメソッドとアクションが解析できる。ユーザ操作トークンは文脈自由文法として扱うことができる。

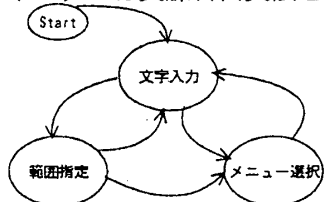


図2 ユーザアプリケーションの状態遷移の例

まずアプリケーションステータスの生成だが、これは図2に示すような有限オートマトンとして捉えることができる。イベント、イベント発生座標、直前のアプリケーションステータスの3つからなるアルファベットを入力し、これをもとに有限のアプリケーションステータスで解析するオートマトンは必ず存在する。もし存在しないならば、アプリケーション内部に問題がある。アプリケーションそのものが、これと同じ機構を組み込んで、内部のアプリケーションステータス遷移を管理しているからである。

ユーザ操作トークンの生成では図3のようにイベント、イベント発生座標、アプリケーションステータスの直積となるようにユーザ操作トークンの生成をするだけなので、これも問題ない。

最後のユーザ測定要因の解析はやや複雑な仕組みになる。イベントとその発生イベント発生座標、発生したときのアプリケーションステータスを基にユーザがどのようなアクションを必要としてどのようなメソッドを行ったか、は文脈自由文法の範囲で解析が可能である。また、そのパーサについては、半自動で生成が可能である。

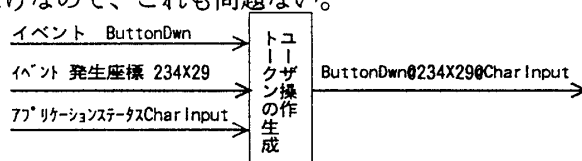


図3 ユーザ操作トークンの生成

このモデルでユーザの行動を解析する場合、解析が可能か否かは主にユーザ操作トークンの種類が何種類かによる。ユーザ操作トークンの種類が多すぎる場合、理論上は解析が可能であっても、実際のコンピュータのメモリ等の都合で実用にはならない場合がある。イベント、アプリケーションステータスについてはユーザ操作トークンの種類は有限であり、一般的にその種類数はそれ程多くない。問題はイベント発生座標についての情報である。

例としてテキストエディタの場合を考える、そのイベント発生座標は非常に少ない。イベント発生座標は離散的に存在し、その密度は均等である。よってイベント、イベント発生座標、アプリケーションステータスによる直積を求めても実用上全く問題なくユーザのメソッドを抽出可能である。

次に行ベースのWYSIWYGワードプロセッサについて考えてみる。ここでいう行ベースのWYSIWYGワードプロセッサとは一般のパーソナルコンピュータ上にあるWYSIWYGなワードプロセッサのほとんどが当てはまる。これらのワードプロセッサの特徴はカーソルの移動は離散的であるが、密度は均等ではないし、一定でもない。一文字一文字がイベント発生座標を指定する最小単位だが、文字の大きさは可変である。しかし、文章の入力フィールドは全体として一本のテープと見なすことができ、解析上はなんら問題はない。問題点として考えられるのは文字幅、文字送り、行送り、が非常に細かく指定できる場合である。この場合、カーソルの位置が具体的にどこを差しているのかを的確に言い当てることはできない(各々のアプリケーションは内部情報でこれらの問題を解決し得る)。このことを考えた場合、行ベースのWYSIWYGワープロの場合、ユーザのメソッドは抽出可能だが、対象アプリケーションの精度を考慮しなければならない。

最後にWYSIWYGのDTP、ドローツールの場合、その解析はかなり制限がある。入力フィールドをテープと見なすことはできないし、一つ一つのオブジェクトのイベント発生座標は取れるがその値は非常に多く(理論上無限に存在する)、このモデルをそのまま用いた場合実用的な構文解析は難しい。

5 考察および今後の課題

本研究で提案したモデルは、知識ベースを作り、ユーザの振る舞いを解析する方法と比較すると知識ベースのルールが必要かつ十分な量だけあるかどうかのチェックしないだけ有利である。知識ベースを作る方法でも対象アプリケーションを忠実に反映すれば提案モデルと同じ機能を提供できるが、実際のルールベースが単なるif thenルールであるために、チェックが難しい。提案モデルではルール解析を文脈自由文法でおさめるようにしてあるのでルールが破綻していないことを調べることができるので有利である。

参考文献

[NAGA93] 長崎、中津、北山、東「個々のユーザに適應したユーザインタフェースの構築 - 3D等高線図によるユーザ習熟度判定 -」(1993経営情報学会春季全国研究発表大会)