

Web 上でのユーザの操作情報を用いた個別指導システムの提案

池端裕子 曾谷俊男

日本アイ・ビー・エム (株) 東京基礎研究所

Web 上での業務アプリケーション (以下 Web 業務 AP) の急速な普及に伴い、ユーザの使用スキル不足をいかに解消してゆくかが今後問題となってくると予測できる。集合研修でスキル習得を行う方法では、ユーザ数が多いため研修コストの負担が大きい。一方、ユーザの多様なスキルレベルに対処するために、ヘルプ機能の埋め込みや詳細説明の付加等 AP 構成に工夫を凝らして開発を行う方法がある。しかし、一つのページに多くの情報を付加すると、AP 開発の規模が増大しコストがかかってしまう。そこで本稿では、開発コストを低く抑えつつ、Web 上でのユーザの操作情報を手がかりに個別指導を自動的に行う個別指導プロトタイプシステム:WebAttendant を提案する。更に、Web 上の研修業務 AP 上に WebAttendant による個別指導サービスを実験的に提供した。その利点、問題点について検討する。

A Tutoring System on Web based Applications based on User's Activity

Yuko IKEHATA, Toshio SOUYA

Tokyo Research Laboratory, IBM Japan

In connection with the rapid spread of the business applications on Web (following Web business AP), how to relieve various levels of skill among users of Web business AP becomes an issue. This paper proposes a tutoring prototype system, called WebAttendant, which features the ability to automatically carry out individual instructions based on the user's Web operation information. This functionality could lower development costs for Web APs. Moreover, this prototype system experimentally was used as a tutoring service on an internal site. The advantages and problems of the system are examined.

1. はじめに

近年、業務アプリケーション(スケジュール管理、財務会計、人事給与、販売管理システム等:以下業務 AP)が広く普及することに伴い、より多くのエンドユーザがこれらのシステムに触れるようになった。

業務 AP を新たに導入すると、エンドユーザは業務 AP の使用方法の理解や業務経験に基づいた応用力(スキル)が必要となってくる。一般にスキル習得には集合研修や同期型の遠隔研修が行われている。しかし、研修だけでスキル習得を使用すると、システムの多様性や更新の頻度があがるに連れ、指導員や受講中の研修生にかかるコストが増加してしまう。そこでユーザに実際の業務を

させながら個別指導を自動的に行うことで、ユーザのスキル向上を目的とした電子的業務補助システム(Electronic Performance Support System:以下 EPSS^[1])の研究が提案されている。

一方、インターネットが社会基盤として認知されるとともに、Web ベースの業務 AP(以下 Web 業務 AP)を導入することが一般になってきた。ユーザが Web 業務 AP を使用して業務目的を達成するためには、少なくとも次のスキルが必要である。

- コンピュータや Web ブラウザの操作スキル
 - 対象とする Web 業務 AP の使用スキル
 - 求める情報がどこにあり、リンク先にどのようなページが存在するかを推測するスキル
- これらのスキルが不足すると、Web ページ上で

何度も同じページを行き来したり、目的達成までに必要以上に時間がかかってしまう。この対処方法として、ヘルプ機能の埋め込みや、詳細説明の付加、サイトマップの提示などアプリケーションの構成に工夫を凝らす方法がある。しかし、多様なユーザのスキルレベルを想定してアプリケーションの開発を行うと、Web ページが煩雑になるだけでなく、アプリケーション開発の規模が増大しコストがかかってしまう。

そこで本稿では、EPSS の概念を Web 業務 AP へ導入し、ユーザの操作を自動検知し解析する。これにより開発コストを低く抑えつつ、個別指導を自動的に行う(以下チュートリング)プロトタイプシステム:WebAttendant を提案する。

2. EPSS の概念

Stevens^[2]らによる EPSS の目的を以下に示す。

- 業務に必要なスキルトレーニングをその場で必要な分だけ提供すること
- 内容や状況に合わせたアドバイスを提供すること

つまり、EPSS とはユーザの状況、仕事内容を考慮して、的確で必要最小限のアドバイスやスキルトレーニングを与えるためのシステムである。

3. WebAttendant の設計目標

EPSS の概念を Web 業務アプリケーションに適応した WebAttendant の設計にあたり、次の 2 つの要件を満たすことを目標とする。

3.1 スキル習得のための時間とコストの低減

Web 業務 AP を使用するユーザ数の増加に伴い、集合研修や遠隔研修では指導員や受講中の研修生にかかるコストが増加してしまう。一方、様々なスキルレベルのユーザを想定して UI を設計すると、UI 構成が複雑化し開発コストも増大する。そこで、システムのコスト削減を目標とする。

3.2 ユーザに負担をかけない個別指導

ユーザがスキルを習得するためには、業務を行

いながら指導ができるチュートリングが有効である。一方、ユーザは業務を達成することが目的なので、チュートリングを受けることが負担となってしまうのは本末転倒である。従ってユーザになるべく負担にならないチュートリングシステムが望まれる。

4 WebAttendant システムの設計

前章の目標を達成するべく、WebAttendant の設計を行った。その概略を以下に説明する。

4.1 業務 AP と WebAttendant の分離

メンテナンスの容易化やメンテナンスコストの低減を狙い、Web 業務 AP とチュートリングシステムを別々にメンテナンスできるようにする。そのため、Web 業務 AP と WebAttendant を分離する。これにより Web 業務 AP 自体を変更することなく WebAttendant サービスを付加や変更できる。

4.2 チュートリング用ルールのシステムからの分離

チュートリング用のルールを他のページでも再利用できるように、ルールモジュールを WebAttendant 実行モジュールから分離する。

4.3 適確なチュートリングの実現

様々なスキルレベルに対応し、ユーザになるべく負担をかけないチュートリングを実現するために必要な機能を以下に述べる。

4.3.1 ユーザの操作情報の自動検知

ユーザになるべく負担をかけないために、ユーザの状況把握の方法は、可能な限り、業務中のユーザの操作情報を自動検知し、この情報からユーザの状況を推測することとする。Web 業務 AP では次の操作情報を取得する機能が必要となる。

- (1) Web 業務 AP では、ユーザは複数のページに跨った操作を要求される。よって本システムは複数のページに跨った操作情報を取得できるようにする。

- (2) Web 業務 AP 上での業務内容が複雑になると、ユーザはページ上で様々な作業を要求される。Web 上でユーザが作業している間、システムはページ上での操作情報を随時取得できるようにする。

4.3.2 チュータリングのための操作情報の解析

ユーザを適確にチュータリングするために、システムはユーザの操作情報を管理することとする。システムはこの中からチュータリングに必要な特定の操作情報を抽出し解析する。

4.3.3 適切な場所への情報提示

適確なチュータリングを行うためには、操作対象となる部分をユーザに示しながら説明をすることが必要となる。WebAttendant は、作業の対象部分の側に必要な情報を提示しながら、チュータリングを行う。

5. 既存 EPSS と WebAttendant の位置付け

WebAttendant は、第 1 章で述べたように、EPSS の考え方を Web 業務 AP に適用することを目指している。既存の EPSS 用ツールの分類と WebAttendant の位置付けを図 1 に示す。

5.1 システム設計の観点からの分類

EPSS はシステム設計の観点から、SI プロジェクトのように対象システムに組み込んでしまう「組み込み型 EPSS」と、対象システムとは分離したツールを使用する「分離型 EPSS」に分類できる。組み込み型 EPSS には、次の問題点がある。

- 対象のシステムと一体化してしまうため、開発やテストが複雑になり、高コストになる。
- 対象システムに特化してしまうと、モジュールの再利用が難しい。

WebAttendant は、業務 AP と WebAttendant を分離しているため、分離型 EPSS に位置づけられる。分離型 EPSS の場合、次の利点がある。

- ツール自身の再利用が容易

	暗黙的手法		明示的手法
	Web ベースアプリケーション向け	ネイティブアプリ向け	一般アプリ向け
分離型 EPSS	WebAttendant	MMhelper Microsoft Agent QuickCards	CoachWare TRACK Knowledge base
組み込み型 EPSS	SI プロジェクト		なし

図1 WebAttendantの位置づけ

他の会社名、製品名およびサービス名等はそれぞれ各社の商標。

- 別々に開発できるため低コストにできる

5.2 ユーザの状況の取得方法による分類

EPSS はユーザの状況の取得方法から次の 2 種類に分類できる。

- 暗黙的手法を用いた EPSS : ユーザが業務作業中に行ったマウス操作や、閲覧に費やした時間等のユーザの挙動から、ユーザの状況に対応する EPSS。
- 明示的手法を用いた EPSS : 知識ベース型ツールを用いて、ユーザに状況説明をさせたり、質疑応答をさせることにより、ユーザの状況に対応する EPSS。

一般に低スキルのユーザはアプリケーションの操作で手一杯な状態になりやすいと思われる。明示型 EPSS では、状況説明や質疑応答を強いるという負荷を、余裕のない状態のユーザにかけてしまう。

WebAttendant では、ユーザの状況把握の手がかりとなる Web 上での操作情報を自動的に検知している。これにより、ユーザになるべく負担にならないチュータリングを実現している。そのため WebAttendant は、暗黙的手法を用いた EPSS に位置づけられる。

暗黙的手法を用いた EPSS の例として他に、Microsoft Agent に代表される製品が挙げられる。これらの製品はネイティブ AP 向けであるため、Web AP 向け EPSS である WebAttendant と区別する。

6. WebAttendant システムの構成

システムの構成図を図 2 に示す。システムは次

のモジュールから構成される。

(1) Proxyサーバ

ProxyサーバはWeb ページ中にイベントプロブ等のモジュール用タグを埋め込む。クライアントには、タグが埋め込まれたWeb ページが読み込まれる^[3]。

(2) イベントプロブ

Web ページ上での CLICK, KEYDOWN 等の DOM イベントを自動検出し、操作履歴情報である操作イベント (DOM イベントの発生時刻、DOM イベントの発生場所、クライアントの IP アドレス、DOM イベント、URL) を取得する。取得した操作イベントはログサーバへ送る^[4]。

(3) ログサーバ

イベントプロブからの操作イベントを受け取り、コンテキストエンジンに送信する。また、ルールエンジンからのガイダンスアクションをガイダンスエンジンに送信する。

(4) コンテキストエンジン

ログサーバから受信した操作イベントを元に、ユーザの操作状況を管理する。必要に応じてコンテキストをルールエンジンに送信する。

(5) ルールエンジン

コンテキストエンジンから受信したコンテキストの内容から、該当するルールを選択する。ルールの内容を解釈し、ガイダンスイベントの発行、あるいはコンテキストエンジンにコンテキストの更新命令を送信する。

(6) ガイダンスエンジン

ガイダンスエンジンは、クライアントアプレットから起動される。ルールエンジンからガイダンスアクションの実行命令を受け取ると、情報提示を行う^[4]。

(7) オーサリングツール

ルール記述を容易にするためのユーザインタフェースを提供する。ユーザはチュートリングの対象となるオブジェクトを直接指定しながらルールを作成できる。

7. 実証実験

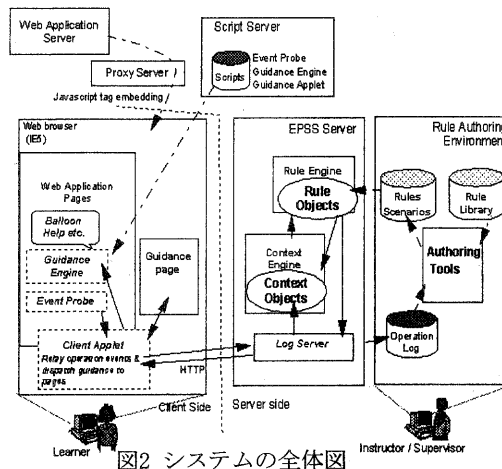


図2 システムの全体図

本稿で提案したWebAttendantによるチュートリングサービスの有効性を検証するために実証実験を行った。今回は社内向けにWeb上で研修の検索、申し込み等を行う研修業務AP上にWebAttendantによるチュートリングサービスを実験的に提供した。

7.1 ルール作成のための事前調査

実験を行うにあたり、Web 業務 AP に必要なルールを作成した。ルール作成には、まず Web 業務 AP のアクセスログをサーバログ解析ソフトを用いて解析した。それに基づき必要なルールについての検討を行った。以下に必要なルールをまとめる。

- (1) ユーザのアクセス頻度が高いページの経路を調査した結果、ID 登録、コース照会、コース申し込み等複数の業務があった。そのため、ユーザがトップページを訪問した時点では、何の目的で来たのかを判断することは困難なことがわかった。そのため、「ユーザの業務目的を把握するためのルール」が必要であると判断した。
- (2) サイトの構成上、「初めて訪問したユーザは ID 登録する必要がある」ということにユーザが気付くことがわかった。そこで「ID 登録の有無をユーザに確認するルール」と、「未登録の場合は ID 登録を促すためのルール」が必要であると判断した。
- (3) 各業務は複数のプロセスからなっている。例

例えば、「コース申し込み業務」の場合、ID登録、コース照会、受講照会、コース申込の4つのプロセスから構成する。業務が多くなると、ユーザは各業務の流れを把握することが容易でない、「業務の流れを把握させるためのルール」が必要であると判断した。

- (4) ユーザが、ID登録の結果を示すページを連続して複数回アクセスしていることが分かった。このことから、ユーザはID登録を何回も失敗している可能性が高いことがわかった。そこで、「業務達成までユーザをナビゲートするためのルール」が必要であると判断した。

7.2 作成したルールの概要

7.1の事前調査から、次の4種類のルールを合計30個作成することとした。次に作成したルールをいくつか示す。ルールは、「条件部」と「実行部」から構成される。条件部の条件が満たされると実行部の記述が実行される、if-then型ルールである。

- (1) ユーザの業務目的を把握するためのルール
ユーザが研修サイトのトップページをロードした後、マウスの移動(mousemove)を頻繁に行う場合、研修APの業務内容を提示し、業務目的を尋ねるウィザードをユーザに提示する。

(条件部) トップページがロードされた時に、ページ上でマウスを5回以上動かした。

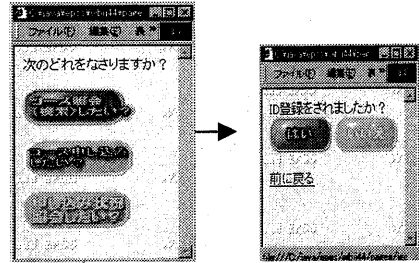
(実行部) 業務目的を尋ねるウィザード (図3参照) をWebページの側に提示する。

- (2) 業務プロセスの流れと次のプロセスを把握させるためのルール

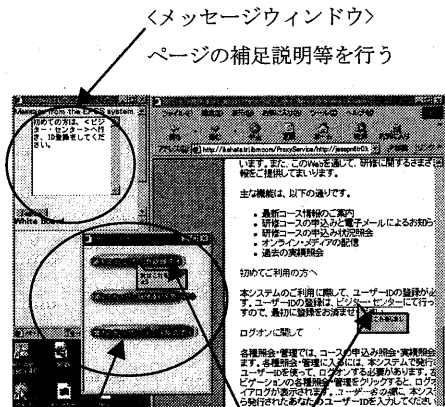
業務プロセス提示ウィンドウを提示し、ユーザが目的とする業務に必要なプロセスの流れと次に必要なプロセスをユーザに把握させるためのルールである。

(条件部) ID登録の有無を質問するウィンドウで「NO」を選択した時 (ID未登録が確認された時)

(実行部) 「次はID登録して下さい」というバルーンヘルプメッセージを、業務プロセス提示ウィンドウ (図3参照) 中のID登録プロセスの側に提示す



<業務目的を尋ねるウィザード> <ID登録の有無を質問するウィザード>



<業務プロセス提示ウィンドウ> <バルーンヘルプ>
次の作業内容や業務に必要なプロセスの流れ図を示す。

図3 チュータリングサービスの概要

- る。
(3) 業務達成までユーザをナビゲートするためのルール

(1)のルールにより、トップページでユーザの業務目的を把握した後、ユーザが業務を達成するまで、ナビゲートするためのルールである。例えば、ユーザがウィザード上でID登録業務を選択した場合、ID登録業務を完了するまで、作業場所を指し示しながら作業内容を指導する。

(条件部) ID登録ページの入力フォーム上にマウスがのった(mouseover)時

(実行部) 「社員番号を半角6桁で入力し、登録ボタンを押して下さい」というバルーンヘルプメッセージが入力フォームを指し示す。

8. 実験結果

実験期間は約1ヶ月である。その間30名の社会人がWebAttendantサービスを利用した。Web上にアンケートを提示し、22名から回答を得た。

アンケートの内容は、これまでに研修サイトを利用した回数、ID登録業務の達成の有無、ID登録以外の業務達成の有無、各業務を困難なくできたかどうか、ユーザが役に立ったルールについてである。

アンケートの結果、初めてのユーザが3名、久しく利用していないため初めて同然のユーザが3名、2-5回使用したユーザが9名、数え切れないほど使用したユーザが7名であった。

初めて訪問したユーザは3名中3名とも「ID登録方法を困難なく見つけ、ID登録ができた」という項目を選択した。よって、初めて訪問したユーザはID登録をスムーズにすることができたことがわかった。初めて訪問したユーザにとって役立つルールは、以下のものであることがわかった。

- ・ ウィザードを用いて業務目的を把握するためのルール
- ・ 未登録者にID登録を促すためのルール
- ・ ID登録業務をナビゲートするためのルール
- ・ 業務プロセス提示ウィンドウを用いて、ID登録のプロセスを把握させるためのルール

上記の結果より、初心者が見つけ、ID登録をスムーズに実行できた理由は以下のように考えられる。

- ・ ユーザは業務プロセスの流れとユーザが次にするプロセスの把握が可能となったために、効率よく業務を行うことができた。
- ・ 作業部分をバルーンヘルプで指し示しながら作業内容が指導されたために、ユーザは間違っていたリンクをたどることが少なくなった。

9. 今後の課題

実証実験を行った際に明らかになった問題点について述べる

(1) ルールの管理

今回の実証実験では約 30 個のルールを作成し

た。作成したルールの数が増えると、各ルールの目的や使用条件の管理が複雑になる。そのため、ルールの目的や使用条件を明確に表示できる管理ツールが必要である。

(2) 動的に生成されたページへの対処

データベースとリンクしページを動的に生成されたページでは、文書構造が一定ではないため、単純にDOMオブジェクトにアクセスすることはできない。Xpointerなどの参照技法を使った場合でもIDを元のサーバで定義しない限り、困難である。ただし、動的に生成されるページに関しては、ある程度ページの構造が予想できるという特性がある。これを利用した参照技法が必要である。

10. 終わりに

本稿では、Web上での自動チュータリングシステムを提案し、そのシステムの概要を述べた。実証実験を行い、システムの利点と問題点について述べた。

謝辞

本論分中で述べた実験には、日本アイ・ビーエム研修サービス株式会社の佐藤武久氏、増田幸恵氏をはじめ、サイトスタッフの方々のご協力をいただいた。ここに謝意を表する。

参考文献

- [1]Gery G, "Electronic Performance Support System", 1991
- [2]Stevens, "Designing electronic performance support tools", improving workplace performance with hypertext, hypermedia, and multimedia.", 1996
- [3]古井陽之助, 青木義則, 土方嘉徳, 「Web ページに操作分析・自動実行プログラムを埋め込むための Web プロキシ」, 第 60 回情報処理学会全国大会 5S-8, March 16, 2000.
- [4] Aoki, Y., Ando, F., and Nakajima, A. (2000). The 7th International Conference on Parallel and Distributed Systems (ICPADS2000), Web Operation Recorder and Player. PP.501-508, July 2000.