キーボードのみで快適な操作が可能な Web ブラウザ UI の設計と実装

豊田 義純 † 佐藤 喬 † 多田 好克 †
電気通信大学 大学院情報システム学研究科 ‡

1. はじめに

一般的な Web ブラウザでは，操作にキーボードとマウスの両方が用いられる．フォームに対して行う文字入力操作にはキーボードが用いられる．一方リンクやプルダウン等，Web コンテンツの選択にはマウスが用いられることが多い．しかし，入力デバイスを頻繁に持ち替える手間は作業効率の低下を招く問題がある．

そこで，キーボードのみで Web ブラウザの操作を可能とする UI を実装した．入力デバイスの持ち替えがなければ，Web ブラウザに対する操作効率の向上が見込める．本研究では一般的な Web ブラウザの例として Firefox を対象とし，Firefox の拡張機能として UI を実装した．

2. Web ブラウザ UI の設計

エディタの操作はキーボード操作と深い関係を持つ．そのため，UI にエディタと似た操作方法を持たせれば，キーボードのみでも Web ブラウザの操作を自然と行える．

また，キーボードのみによる Web ブラウザ操作の効率化を図るために考慮すべき点は以下のように大きく二点挙げられる．

1. フォームなどへ文字を入力する場合とキーボードによって Web ブラウザの操作を行う場合とを分けて考える必要がある．
2. マウスによる操作がキーボードのみでも実行可能であり，かつユーザに対し視覚的なフィードバックを与える必要がある．

上記の二点について，それぞれ次節以降に述べる．

2.1. Web ブラウザ操作の種別分け

Web ブラウザの操作には，Web コンテンツを対象とするものと Web ブラウザを対象とするものとがある．これらが持つ意味合いは異なるため，種別を分ける必要がある．例えば，クリックは Web コンテンツに対する操作であるが，閲覧履歴を一つ戻す “戻る” は Web コンテンツではなく Web ブラウザを対象とする操作である．前者は閲覧している html の内容を考慮するのに対し，後者は考慮していないという違いがある．

Web コンテンツを対象とする操作も二つの種別に分ける必要がある．例えば，カーソルの操作をキーボードで行う時，これをフォームへの文字入力とは異なる種別のものとして考えないと，フォームへは意図しない文字入力がなされてしまう．

種別分けされた各操作は互いに排反であるため，それぞれ排他的に扱うことができる．しかし Web ブラウザを対象とする操作は，常に実行される可能性がある．そのため，これらの操作はいつでも実行可能か，あるいはあまり手間をかけずに実行可能とする必要がある．

種別分けされた Web ブラウザ操作を図 1 に示す．

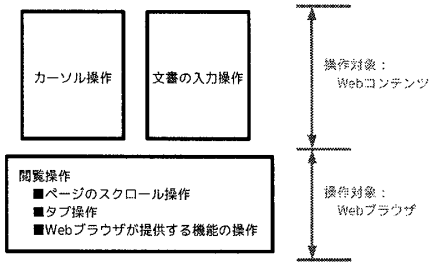

図 1: Web ブラウザ操作の種別分け

2.2. グラフィクスカーソルの重要性

通常，OS が提供するグラフィクスカーソルをマウス等のポインティングデバイスを用いて操作することで，ユーザは Web コンテンツの選択を行っている．しかし，その操作もキーボードのみによって実現できなければならない．

通常 Web ブラウザは，クリッカブルな Web コンテンツを Tab キーによって順番に巡る機能や，Enter キーによって Web コンテンツを選択する機能を提供している．この機能を用いれば，キーボードのみでも選択操作が行える．また，クリッカブルな Web コンテンツを一意にラベリングし，ラベルのキータイプとラベルに対応するコンテンツの選択とを関連付ける方法 [3][5] も存在する．しかし，これらの方法では Web コンテンツの選択にグラフィクスカーソルを用いないため，操作には違和感が残る．

Web コンテンツの選択はキーボードのみで可能で，かつグラフィクスカーソルを用いて実現することが好ましい．つまり，キーボードによってグラフィクスカーソルを操作する必要がある．

3. Web ブラウザ UI の実装

GUI 操作が可能であり，かつ一般的な Web ブラウザの例として本研究は Firefox 3（以降，Firefox）を対象とした．また，UI は Firefox の拡張機能として実装した．

拡張機能はディスパッチャとディスパッチテーブルから構成されており，Web ブラウザの操作はディスパッチテーブルに記述されている．なお，拡張機能が持つディスパッチテーブルは複数である．この点については 3.1. 節に後述する．

Firefox に対してユーザがキー入力を行うと，キーイベントが発行される．このキーイベントは拡張機能が受け取る．受け取ったキーイベントがディスパッチテーブルに記述されている場合にのみ，ディスパッチャが Firefox に対し，キーイベントに関連付けられたコマンドを発行する (図 2)．

また 2. 節で挙げた二点に対する実装について，それぞれ次節以降に述べる．

3.1. 操作種別モードの実装

Web ブラウザの操作を行う際には，その操作の種別を考慮しなければならない．ここで，vim が複数のモードを切り替

†Tomoyoshi Toyoda, Takashi Satou, Yoshikatsu Tada
‡Graduate School of Information Systems, The University of Electro-Communications

えながら操作を行うエディタであることに着目した．拡張機能は，Web ブラウザ操作の種別ごとにモードを設定し，複数のモードを切り替えながら Web ブラウザの操作を行う．これにより，vim の操作に似た感覚で Web ブラウザの操作が行える．

モードには，それぞれ内容の異なるディスパッチテーブルを持たせた．モードの切り替えと同時に，ディスパッチテーブルも対応するものへと変更する．これにより，モードの切り替えと実行可能な操作の変更とを関連付けた．なお 2.1. 節で Web ブラウザを対象とする操作として述べたもののいくつかは，特別なディスパッチテーブルに記述した．ここに記述された操作はモードに依存せず，常に実行可能である．

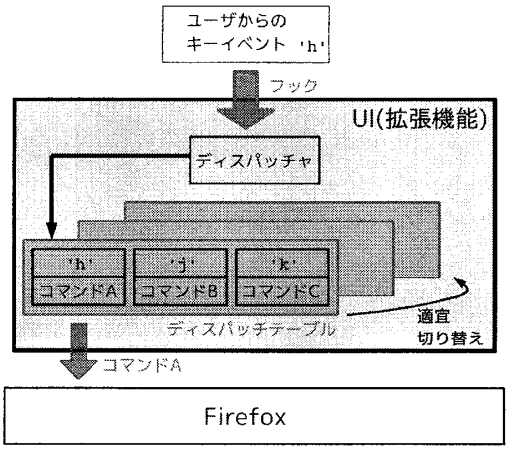

図 2: キーイベントのフックと，追加機能の実行

3.2. 仮想カーソルの実装

キーボードだけでグラフィクスカーソルを操作する必要がある．しかし，Web ブラウザは OS が提供するグラフィクスカーソルを制御できない．そこで，Web コンテンツの選択を専門に扱うようなグラフィクスカーソルを実装した．このカーソルは OS が持つカーソルとは異なるため，区別のため仮想カーソルと呼ぶことにする．

仮想カーソルは，Web ブラウザが読み込む html に対して拡張機能が自動的に埋め込むものである．これにより Web ブラウザ上に表示されているように見える．仮想カーソルの実体は html の SPAN 要素であり，その内部に IMG 要素を持つ．

仮想カーソルの移動は，その SPAN 要素の表示座標を動的に変更することによって実現した．また，仮想カーソルの表示座標に対し，マウスイベントをシミュレートすることによって擬似的なクリック操作を可能にした．これらの制御も全てキーボードのみで行う．

仮想カーソルに求められる，Web コンテンツの選択のために必要な操作性を次節以降に述べる．

3.2.1. Web コンテンツの表示座標の利用

範囲選択操作を除けば，クリッカブルな Web コンテンツのみが選択される可能性を持つ．そこで Web コンテンツの選択を効率的に行うために，クリッカブルなもののみを選択対象とする．そのため，Web ブラウザの画面内における，Web コンテンツの表示座標情報を利用する．このとき，仮想カーソルはクリッカブルな Web コンテンツが表示されている場所のみを順番に回る．

3.2.2. 再帰的な画面分割を用いた，Web コンテンツの選択にかかる手間の軽減

クリッカブルな Web コンテンツを多く含む html を閲覧する場合，前節 3.2.1. の方法だけでは，仮想カーソルの現在位置から離れた場所にある Web コンテンツの選択に手間がかかる．そこで，Web ブラウザの画面領域を再帰的に分割し，選択の可能性がある Web コンテンツの絞り込みを行う．ある領域が選択された時，その領域内に存在する Web コンテンツのみに選択の可能性を残すものとすれば，選択の手間が軽減できる．マウスキーに対する先行研究 [2] を考慮し，本研究では画面領域を 3 × 3 に分割する．

4. 関連研究

w3m や Lynx のようなテキストブラウザの操作は全てキーボードのみで行える．しかし，現在多くの Web ページは GUI 操作が可能な Web ブラウザ向けに作成されているため，テキストブラウザでは表示できない Web コンテンツがある．しかし，Firefox を対象とすればその問題を解決できる．

TrackPoint 等のポインティングデバイスや視線入力インタフェース [1] のような専用のハードウェアによっても問題は解決するが，Web ブラウザの操作のためだけに導入コストが高い．一方拡張として実装すれば，Web ブラウザを限定するのみであり，導入コストが低く済む．

Firemacs[4] や Vimperator[5] はともにエディタに似た振る舞いを Web ブラウザに持たせるものである．前者は Web コンテンツの選択について言及しておらず，後者はグラフィクスカーソルを用いずに Web コンテンツの選択を行うものである．Web コンテンツの選択にグラフィクスカーソルを用いることで，分かりやすい選択操作が可能になる．

5. おわりに

本研究は Web ブラウザの操作の効率化を図るべく，キーボードのみによる操作が可能な UI を実装した．これにより，マウスとキーボードとを持ち替えることなく，Web ブラウザを操作できる．また，グラフィクスカーソルを用いることにより，Web コンテンツの選択を行う際に視覚的フィードバックが得られる．加えて，クリッカブルな Web コンテンツの表示座標を考慮することにより，効率的な選択操作が行える．

今後は基本的な Web ブラウザの操作に対象とし，本 UI を用いる場合と従来どおりマウスを用いる場合とで操作に必要となる時間を比較することにより，本 UI の有用性を検証する．

参考文献

[1] 久野悦章，八木透，藤井一幸，古賀一男，内川嘉樹：EOG を用いた視線入力インタフェースの開発，情報処理学会論文誌 Vol.39, No.5, pp.1455–1462.

[2] 遠藤光，伊藤久祥，伊藤憲三：再帰的画面分割を用いた肢体不自由者向けポインティング補助手法の開発と評価，IEICE Technical Report, WIT 2007-103, pp.73–78.

[3] Hit-a-Hint, `http://hah.mozdev.org`

[4] Firemacs, `http://www.mew.org/~kazu/proj/firemacs/`

[5] Vimperator, `http://vimperator.org`