LMS インターフェースの簡易作成装置の開発

石川　貴彦, 赤間　清

名寄市立大学 保健福祉学部

北海道大学 情報基盤センター

ishikawata@nayoro.ac.jp

概要：LMS（学習管理システム）のように、Web 上で操作するシステムを開発する場合、その大半は HTML の FORM や INPUT タグを記述して Web ユーザインターフェースを作り上げていく過程をとる。これらのタグには追加属性があり、その属性の細かさが開発者にとって複雑さや面倒さを与えている。そこで本研究では、開発者もしくは教師がインターフェースを容易に作成でき、そのまま１つの操作画面として利用できる装置を開発した。そして、この装置を試用し、開発・検証の効率化や装置活用の将来性について論じた。

1　はじめに

Web ベースシステムの多くはデータベース（以下 DB）と連動しており、Web 上で入力した情報を DB に書き込むことと、DB から必要な情報を選択して Web 上に表示することの繰り返しで、画面を動的に更新していく。そのようなシステムの開発において、Web ユーザインターフェースは設計・製作の頻度が高く、システムそのものの使いやすさを左右するポイントになる。インターフェースの作成は通常、HTML の FORM や INPUT タグを用いて行われ、各種タグは様々な追加属性を持つ。それら属性の記述により、フォームを詳細にコントロールできる反面、開発者には細かな記述が強いられ、それが複雑さや面倒さを与えている状況にあると予測される。

このようなインターフェース作成の面倒さといった問題に対して、これまで様々な解決策が講じられてきた。その例として、GUI 操作でフォームの構成要素をはめ込み、文字などを後から追記していくような Formbakery[1]等の装置がある。この手のシステムは、GUI ではめ込んだ画面から HTML を出力し、それをコピーして自分の Web ページに貼り付ける方法を採り、HTML 生成器という位置づけと捉えられる。DB と接続する場合、単なるソースのペーストだけでは CGI は実行できず、データ送信先のパスや隠し項目の設定など、さらに細かい記述を必要とする。

そこで本研究では、インターフェースの簡易作成だけに留まらず、作成した画面がそのまま CGI 実行できるような装置を開発し、開発や検証の効率化、装置活用の将来性について考察することを目的とした。本装置により、表示される画面や DB アクセスの挙動を、稼動中のシステム上で直接確認するので、作成したインターフェースを、そのままシステムに投入できることが期待される。

2　作成装置開発の前提

2.1 S 式（アトム）による CGI の記述

まず、本研究の Web ベースシステムは、2001 年から筆者らによって独自開発されたものであり、主に LMS として利用している。本システムは、ルール型言語の ET 言語[2]を用いて CGI プログラムを記述していることが特徴であり、述語と引数の列からなる S 式（アトム）が構成要素となる。なお、引数がない場合は述語だけのアトムになる。

（述語　引数 1　引数 2　…）

（述語）［引数がない場合］

アトムは、ユーザが独自に述語を定義できる D アトムと、ET 言語のビルトインである B アトムの２種類を用いる。CGI プログラムは、これらのアトムを並べて作られるが、D アトムの多くは、引数の値を printf アトム（B アトム）に代入して、HTML タグを発行している。以下は CGI 用に定義された D アトムの一例であり、上から順にテキストフォーム、ラジオボタン、セレクトボックスを表している。

```
(form:text "name" 20)
(form:radio "grade" (("1" "1 年") ("2" "2 年")))
(form:select "faculty" (("L" "文学部") ("E" "工学部")))
```

各アトムの第１引数には name 属性が入り、第２引数には value 属性が入る。ただし、ラジオボタンやセレクトボックスは選択肢になることから、この場合の第２引数は、value 属性と表示項目のペアからなるリスト列を記述する。HTML でラジオボタンを書く場合は、

```
<INPUT type="radio" name="grade" value="1">1 年
<INPUT type="radio" name="grade" value="2">2 年
```

と記述量が多くなり、開発者に複雑さをもたらす要因になっている。それらをアトムやリスト列で表現することで、書き方が集約され、開発者に直感的な理解を与える。

そして、INPUT タグの他に、TABLE タグ等についても D アトムで定義した。以下に例を示す。

```
(table) / (table (("width" 200)("height" 65)))
(td) / (td (("align" "center")))
```

これらは引数がないアトム（前部）であり、HTML の<table>や<td>と同意である。したがって、D アトムでわざわざ定義する必要性は薄いように感じるかもしれない。しかし、table や td タグにも追加属性はあり、前述の例と同様、引数に属性のリスト列（後部）を記述することで、全体の統一感が得られると判断した。

2.2 Web ベースシステムのフレームワーク

本システムは Moodle と同じく、機能毎に分化したサブシステムを、コアシステム上に搭載して運用するという機能追加型のシステムになっている。本システムの場合は、サブシステム単位で LMS 全体を構成するだけでなく、サブシステム中に含まれる１機能を最小構成単位としているので、様々な機能を組み合わせて、独自の機能群として構成できるという特徴を持つ。そのため、Moodle よりも構成の幅が広く、自由度の高いシステムを構築できる環境にある。

また本システムは、ユーザ管理、コース管理、メニュー管理、システム管理の４つの管理機能を基本とし、前述したサブシステムの開発・追加によって運用する。このように、本システム自体が、機能の開発や拡張を容易に行える枠組みを保証している。今回開発した簡易作成装置は、システム管理の機能の１つとして試作し、稼働中のシステムに搭載した。

3 インターフェース簡易作成装置

3.1 入力画面

簡易性を高めることを優先するため、インターフェースの作成は、Web 上にある各フォームに対応したボタンをクリックして行われるようにした。なお、枠内に直接 D アトムを書くことも可能である。さらに、ボタンのクリックで入力されたアトムは、引数のない D アトムを極力表示した（図１）。これは、最初から属性を考えないで、大まかにフレームワークを作ることを意識させるようにして、仕上がるインターフェースの見通しを、ユーザに直感的に示すためである。

3.2 D アトムの変換

次に、ボタンから入力した D アトムを変換し、属性を付与した引数付きの D アトム（以下、属性 D アトムと呼ぶ）を提示する。これは、図１の変

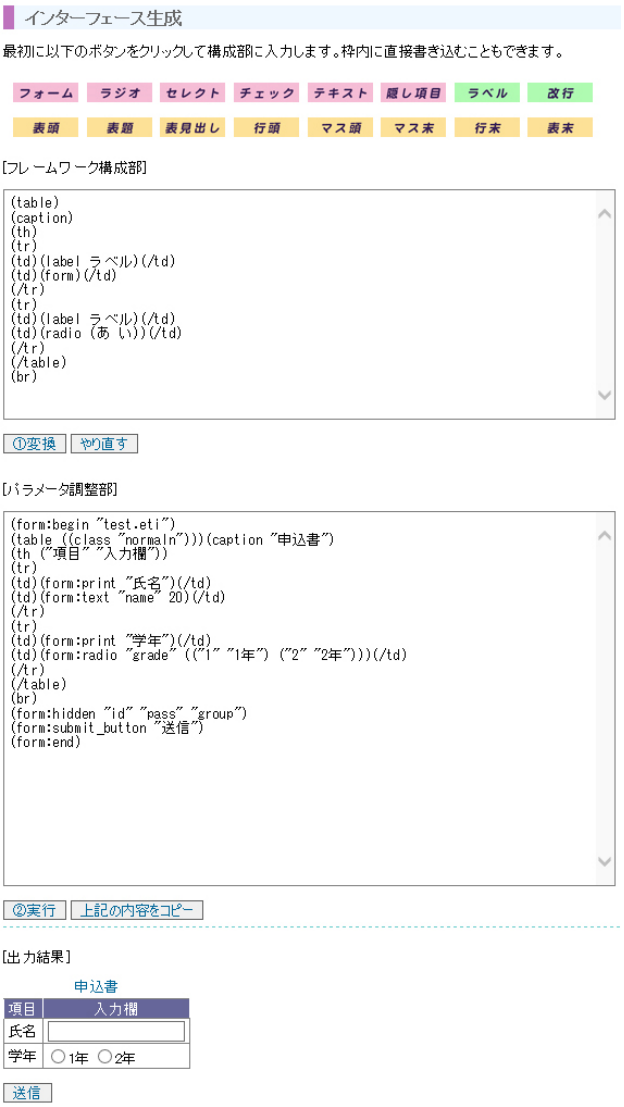

図１　インターフェース簡易作成装置

換ボタンを押すと、変換結果が直下に提示される仕組みになっている。変換した際、引数にはダミーデータを与えており、ユーザはダミーデータの箇所を、実際の属性値に修正するだけでよい。

このように、ボタン入力でフレームワークを大まかに作るフェーズ(フレームワーク構成部)と、引数の値の書き換えによって、属性のパラメータを調整するフェーズ（パラメータ調整部）に分けることで、インターフェースの作成を簡易かつ段階的に行える環境を構築した。

3.3 インターフェースの作成

属性 D アトムを図 1 のボタンで実行すると、稼動システムの CGI プログラムを適用し、実際に使用するインターフェースそのものが提示される。これはパラメータ調整部の直下に作成されるので、これまでの D アトムとの対応が、1 つの画面上でまとめて見て取れる。したがって、属性を徐々に調整しながら、出来上がる画面を逐次確認すればよいので、対話的にインターフェースを作成することができる。このことが、ユーザに開発の容易さを感じさせる大きな要因となるだろう。例えば、WordPress のフォーム生成プラグインの 1 つである MW WP Form[3]は、本装置と同様、ボタン入力でユーザの開発負担を軽減している。しかしながら、そのボタン入力されたフォームが、どのような画面に仕上がっているのかを、同一画面上で直接確認できない。このような別画面表示は、Web ベースのフォーム生成装置に散見される仕様であり、本装置ではそれを克服している。

4 機能の試用と将来性

4.1 開発効率の向上に関する検討

本装置を用いることによる開発効率の向上について、いくつかのフォームの作成から検討した。図 1 のフレームワーク構成部に記述された D アトム列は、その文字数をカウントすると 123 文字で、次のパラメータ調整部の属性 D アトム列は 296 文字であった。そして、最終的に作られたインターフェースの HTML ソースは 624 字であった。通常の方法で直接 HTML を書く場合、ユーザはその 600 字程度のキー入力が求められる。しかし、本装置を用いれば、D アトム列の 123 文字は全てボタン入力で挿入され、変換後の属性 D アトム列 296 字も、大半が自動生成されたものである。なお 296 文字中 50 文字分が、ユーザ自身で属性を修正した箇所であった。つまり、ユーザがキー入力する量は、HTML 直書きの場合と、本装置で作成した場合とで単純比較すると、600 字程度から 50 字程度へと 10 分の 1 程度に縮減できた結果となった。このキー入力量の低減が、作業時間の短縮やソースコードの見渡しやすさにつながり、結果的に開発効率が上がると推測される。

4.2 動作検証に関する検討

本装置で作成したインターフェースは、稼働中のシステムを直接操作できるので、適切なパラメータを与えれば、DB に格納されている情報を表示させることができる。例えば、動画配信サブシステムの操作画面を作成し、その画面から動画一覧を提示させることを考える。本装置の場合、図 2 に示した通り、属性 D アトム列と出力されたインターフェースが同一画面上にあるので、画面作成と動作検証を連続的に行うことができる。動作検証の例として、どの動画一覧を提示するかを確認する作業を取り上げる。ユーザは、インターフェースからプルダウンメニューで一覧を選び、続

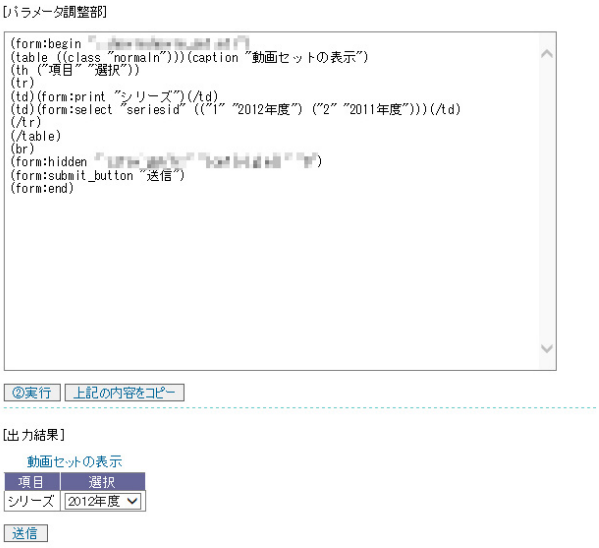

図 2　本装置から作成した動画配信サブシステムのインターフェース

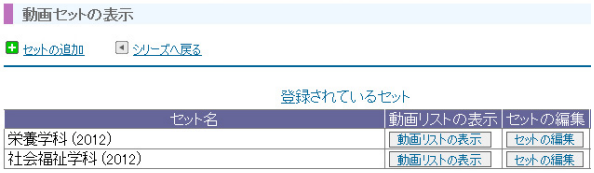

図 3　インターフェース操作後の提示画面

けて送信ボタンをクリックする。図３の一覧が示されれば、属性 D アトム列は、完成されたインターフェースとしてそのまま利用できる。もし、CGI エラー等を引き起こした場合は、属性 D アトムを編集・変換して再度検証すればよい。

このように、インターフェースの作成が動作検証込みで行えるので、作成と検証のプロセスを短期的かつ反復的にできることが、ユーザに開発のしやすさを与えている。

4.3 本装置の将来性に関する検討

本装置の将来的な活用方法として、ピンポイントの DB アクセスに適したインターフェースを、ユーザが手元で作って操作するという方法を提案する。多くの LMS などは、システムが階層構造になっているので、リンク先の深い部分で設定する場面が多い。図４上部は、教師が動画一覧を選択し、一覧に含まれる１つの動画を選んで再生するという流れを示している。もし、１つの動画しか再生しないことが決まっているなら、図４中部のように、本装置でそれのみを再生するインターフェースを作成すればよく、ユーザが毎回一覧から動画を選択するという手間を省くことができる。同様に、図４下部は開講コースから名簿を選んで編集するという流れであるが、これについても、そのコースに特化した名簿編集機能を本装置で作成すれば、最初にコースを選択するという手間を省くことができる。

またパラメータ調整部は、属性 D アトムの入力装置であるため、変換によって表示されたインターフェースは、アトムの実行結果の一例に過ぎない。つまり、システム内に定義された D アトムを、本装置から実行できるので、例えば、２つのリストを結合したリストを求める append アトムが定義されていれば、装置の画面上で解を求めることができる（図５）。したがって、定義済の D アトムがシステム内に存在していることを、装置で確認してから実装すれば、冗長な D アトムの定義を事前に防止することができる。

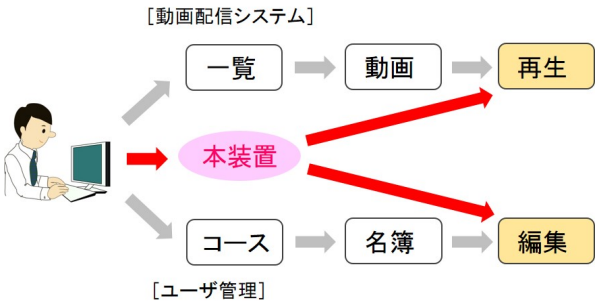

図４　本装置を用いた直接操作

図５　定義済 D アトムの確認

5 まとめ

本研究では、Web ユーザインターフェースの簡易作成と同時に CGI 実行できるような装置を開発し、いくつかの例をもとに、開発や検証の効率化、装置活用の将来性について考察した。

開発した装置は、D アトムの配置と、属性値になる引数の付与によって、直感的かつ少ない記述量で様々なインターフェースの作成を可能にした。また、作成した画面を稼働中のシステムと連動したことで、作成と検証の作業を一貫して行える環境を構築できた。しかしながら、直接実行できる環境が誤操作でもシステムに反映し、セキュリティを脆弱にする問題がある。例えば、検証中に誤ってテストデータを DB に書き込んだり、悪意のあるユーザが DB を削除したりする可能性を否定できない。したがって、これらの対策を熟考して、今後は装置の改良を図っていく。本研究は、平成 25 年度北海道大学情報基盤センター共同研究および科学研究費補助金若手研究 B（課題番号 24700903）の助成を受けたものである。

参考文献

[1] Formbakery、http://formbakery.com/
[2] 赤間清、繁田良則、宮本衛市、「論理プログラムの等価変換による問題解決の枠組」、人工知能学会誌、Vol.12、No.2、pp.266-275、1997
[3] MW WP Form、http://2inc.org/manual-mw-wp-form/