

Dynamic HTML を用いた物理現象シミュレータの製作

村山芳幸¹⁾、杉本一樹²⁾、谷口和成³⁾

1) 京都教育大学教育学部 理系教育学科 rik91144@kyokyo-u.ac.jp

2) 京都教育大学教育学部 産業技術科学科 san92089@kyokyo-u.ac.jp

3) 京都教育大学教育学部 理学科 guchi@kyokyo-u.ac.jp

キーワード：シミュレータ、Dynamic HTML、Java

(受付け：2002 年 4 月 15 日)

I. はじめに

現実の世界における物理現象は、いくつもの要因が複雑に絡み合い、それらの相互作用の結果として生じている。それらの振る舞いを調べるための有効な手段のひとつとして、シミュレーションが挙げられる。シミュレーションとは、「ある物理系の振る舞いを調べるために、そのいくつかの特徴を別の系の振る舞いによって表現すること」(西川、1992)と定義される。つまり、複雑すぎる現象の中から着目したい特徴を抽出し、現象を単純化することによって、その本質に迫る探求手法ともいえる。一般に、その現象を数学モデルにより表現し、コンピュータを用いて解析することにより必要なデータを収集する手法が採られ、そのためのソフトウェアはシミュレータと呼ばれる。

近年、コンピュータの高性能化に伴い、多種多様なニーズに合わせたプログラム言語が開発され、様々な物理系の振る舞いをコンピュータ上で視覚的に再現・解析するシミュレータが比較的容易に作成できるようになった(高嶋、2000)。そのため現在では、いわゆる研究者のみならず、現場の教師によっても作成され、個人の Web 上で公開したり、実際に授業で用いられたりしている。シミュレータが授業に用いられる理由には、実験の補佐およびその理解のための補助的な役割が大きい。一方で、探求活動の一環として生徒に実際に実験をさせることが重要であると知りつつも、昨今の授業時間数の削減等によって、なかなか実験のための時間をとることができないという教師のジレンマもその使用理由として少なからずあると思われる。また、設備や実験上の安全性の問題により実験を行うことが困難な場合もある。いずれにせよ、現場の教師が授業に用いるために忙しい合間を縫ってシミュレータを製作することは、科学教育にかける情熱がなせる業であろう。

しかしながら、一般に流布している Web 技術 (HTML: Web 用の言語や Java: プログラミング言語など) を用いてシミュレータを製作する場合、物理における様々な分野を網羅しようとすればするほど、その作成プロセスは煩雑なものになり、膨大な製作時間がかかる。このことは忙しい現場の教師にとって負荷となることが考えられる。そこで、本稿ではシミュレータ製作における煩雑なプロセスから解放すべく、今回製作した「物理現象シミュレータ」を例に、シミュレータ製作に有効な Web 技術を紹介する。

II. 製作に用いた Web 技術について

Web 技術について、関連する部分のみを簡単に説明する。詳細については、文献（例えば、岡蔵、1999）を参照されたい。

1. Java

Java は 1995 年に Sun Microsystems 社により、ネットワークを意識して開発されたプログラミング言語。現在、世界中の多くの個人および企業ユーザによって幅広く活用されている。代表的な特徴は次の通りである。

- ・ 『Java 仮想マシン (Java Virtual Machine)』と呼ばれるソフトウェアを組み込むだけで、Microsoft Windows、Macintosh OS などのプラットフォームの種類に依存することなく同様の結果を得ることができる。
- ・ 『Java アプレット』という Java プログラムを作成することによって、Web ブラウザに取り込み、インターネットを介しての利用や、動的な表現が可能になる。今回製作したシミュレータにおける個々のシミュレーションのプログラムは、Java アプレットを用いている。

2. Java スクリプト

Web ページを記述する言語 HTML (Hyper Text Markup Language) では実現できない、Web 上における動的な表現を可能にするスクリプト言語 (簡易プログラミング言語)。

3. スタイルシート

HTML は一般に文書の論理構造やレイアウトなど画像表示を指定するために使用されるが、その内の画像表示に関する指定を専門に行う文書の雛形ファイルのこと。Web 上で使用する際には CSS という言語を用いて、HTML 文書から画像情報などを分離した CSS ファイルを作成し、適用することにより画像の統一やプログラムの簡略化が可能となる。

4. Dynamic HTML

上記 2、3 を HTML 文書の中に埋め込むことにより、HTML では実現できない動的な表現を可能にした Web 技術。この技術を用いることにより、プログラム構造が単純化 (階層化) され、画面の更新 (再読み込み) をすることなく画像を切り替えることが可能となる。より詳細な特徴は次節にて紹介する。

III. 製作した物理現象シミュレータ

1. 画面構成と操作方法

製作した物理現象シミュレータの画面と画面構造の概略を図 1 に示す。画面は、タイトル、メニュー、シミュレーション実行の 3 つの画面から構成され、それぞれ①タイトルファイル、②メニューファイル、③プログラムファイルによって制御している。また、それぞれの画面のレイアウトはスタイルシートによって定義している。

シミュレーションの実行方法は次のとおりである。はじめに、図 2 に示されるメニュー画面において物理学における各 [分野] (図 2(a)) の中から該当する分野を選択する。次に、選択した分野の隣に表示される [カテゴリ] (図 2(b)) の中から実行したいシミュレーションを選択する。図 2 では「電磁気学」が選択された結果、その隣に「自己誘導、等電位線、…」などのカテゴリが表示されている様子がわかる。メニューファイルで選択されたカテゴリは対応するプログラムファイルにリンクされており、選択されたシミュレーションは画面

上③の領域で実行される。シミュレーションごとに適当なパラメータを入力して、[実行] ボタンをクリックすると、その場に結果が表示される。



図1 画面構成（実際の画面（左）とその構造（右））

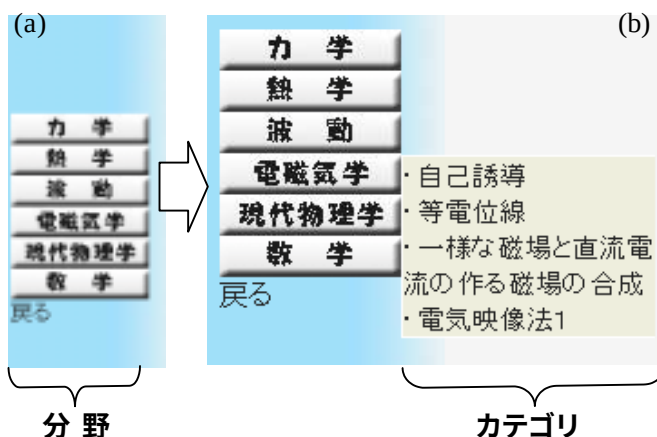


図2 メニュー画面の拡大図

2. プログラム構造

本シミュレータのプログラム構造の特徴を示すために、一般に流布している HTML のみで本シミュレータを作成した場合と、Dynamic HTML を用いて作成した場合の特徴とプログラム構造を比較する。HTML のみで作成した場合、メニュー画面においてある分野を選択し、含まれるカテゴリを表示するためには、分野ごとにリンクされたカテゴリ画像（図 2(b)）を含む画面全体（一画面分）の情報を読み込む必要がある。すなわち、メニュー画面のみの更新はできない。さらに、表示されたカテゴリの中からあるシミュレーションを実行するにも、カテゴリにリンクされた実行画像を含む画面全体を読み込むことになる。つまり、分野およびそのカテゴリを選択するごとに画面情報を再読み込みしなくてはならず、このプロセスはコンピュータに負荷をかけ、読み込みに要する時間も無視できない。一方、Dynamic HTML を用いると、レイアウトを制御するスタイルシートと簡単なプログラムを記述することが可能な Java スクリプトを HTML 文書に埋め込むことが可能である

ため、画面全体の更新をすることなくスムーズに次の画像に移ることが可能となる。

次に、プログラム構造に着目する。メニュー画面において [分野] を選択すると、そこに含まれる [カテゴリ] が表示され、その中から選択してプログラムを実行させるという構造を階層構造という。本シミュレータにおいては [分野] を第 1 層、[カテゴリ] を第 2 層とする構造となっている。図 3 は

HTML のみで作成した場合のプログラム構造を示している。図に示されるように、それぞれ階層構造を持つ [力学分野ファイル] と [電磁気学分野ファイル] が独立したファイルとして並べられ、それぞれのファイルの中に互いの階層構造 (ファイル) を持つという構造になる。つまり、[分野] の数だけ独立したファイルが存在し、お互いがリンクしあうという構造である。このような構造の場合、

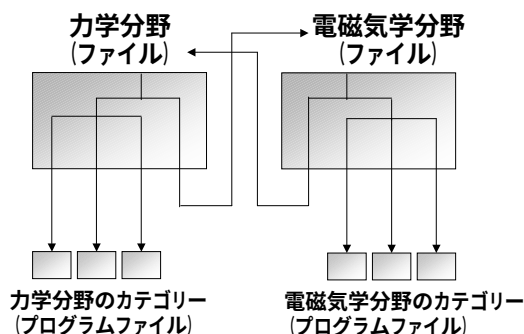


図3 プログラムの構造 (HTML のみ)

分野およびそのカテゴリを選択するごとにリンク先に用意された画像を読み込み直す必要があります、その製作過程において非常に煩雑な作業が必要となる。例えば、図3の構造を持つWebページに、あらたに階層構造を持つ [A] という分野を追加する場合、図4に示すように、互いをリンクするためそれぞれのメニューファイルに [A 分野ファイル] を追加しなければならない。この作業は、[分野] が増えるほど煩雑になり、それに伴い製作のための時間も労力も膨大なものになることを意味する。一方、Dynamic HTML を用いるとこの問題は解決される。図5にその構造を示

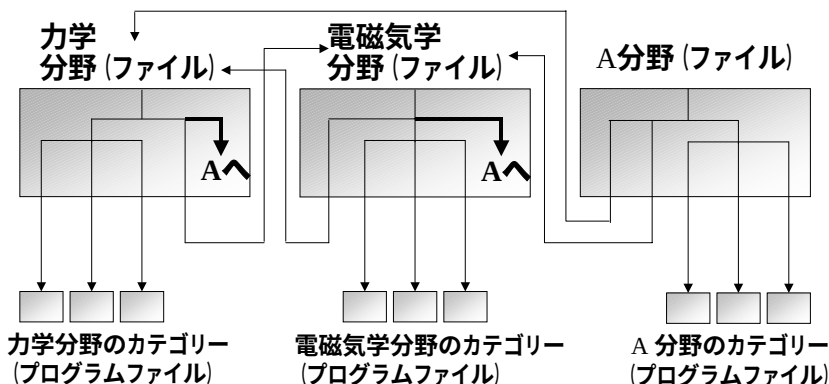


図4 プログラムの構造 (A 分野追加、HTML のみ)

す。メニュー部分に Dynamic HTML を用いると、すべての[分野]ファイルを1つのファイルとしてまとめることが可能になる。したがって、先ほどと同様に [A 分野] を追加する場合においても、A 分野の階層構造を持つ [A ファイル] を一体化されたメニューファイルに追加するのみでよい。ソフトウェアである以上、初めから完璧なものを製作すること

は不可能であるため、修正や追加などのプログラムの更新作業を行う際に非常に有効な技術となる。

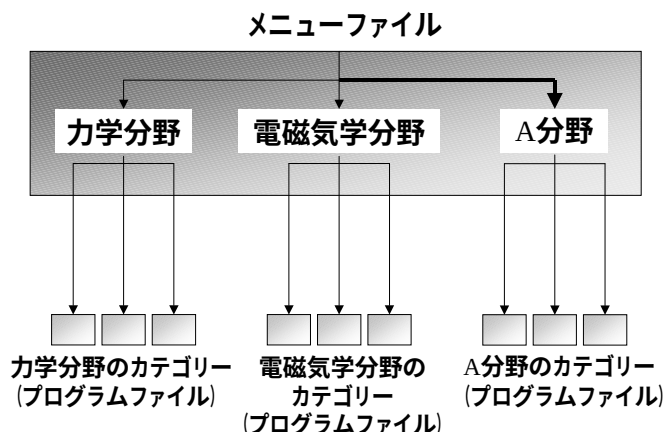


図5 プログラムの構造 (A 分野追加、Dynamic HTML 使用)

今回作成した物理シミュレータは、授業時の補助教材として用いることを目的としているため、教育現場に配備されているパソコン上で問題なく動作する必要がある。本シミュレータのメインプログラムは Java を用いているため OS の種類に依らず、本シミュレータをホストサーバにインストールするだけで Web を介した授業が可能である。しかしながら、Dynamic HTML は、最近の一般的なブラウザ (Microsoft Internet Explore や Netscape Navigator など) には標準搭載されているものの、その仕様に互換性がない部分がある。現状では、Microsoft Windows では Internet Explorer ver.5.0 以上、Netscape navigator ver.4.75 以上においてシミュレータの正常動作を確認しているが、今後はブラウザの種類やバージョンを判別し対応させるプログラムを追加するなどして、ブラウザの種類に依存することなく動作させることが課題である。

3. 製作したシミュレータの実行例

製作した物理現象シミュレータの典型的な実行例を示す。ここで、物理現象のプログラムは、「新／BASIC による物理」(平田、1988)を参考に作成した。図6は [分野:電磁気学] → [カテゴリ:自己誘導] を選択した場合に表示される画面全体の様子を示している。ブラウザ上 (図中※) にパラメータ (ここでは電源電圧など) を直接入力し、[実行] ボタンをクリックするごとに出力画面のみが更新・描画されるため、様々な例を瞬時に効率よく表示することができる。



図6 シミュレータの実行例

IV. まとめ

Dynamic HTML を用いて、物理現象シミュレータを製作した。この技術を用いることにより、シミュレーションの選択・実行の際に画面が再更新されないため、コンピュータにかかる負荷が低減される。さらに、シミュレータのプログラム構造が単純な階層構造となるため、製作に要する時間やプログラムを更新する際の煩雑性が大幅に削減されることが期待される。科学教育において可能な限り実験・観察を中心とした探求活動を行うことが重要であるが、現象の理解のための補助的な役割としてのシミュレータの果たす役割は少なからずあると思う。使用にあたっては、仮想的（限定条件下）であるということと現実との対応を生徒に十分に認識させることが重要であるが、特に、準備のための時間や安全性の問題で実際に実験を行うことが困難な現象に対しては、生徒自身が実際に様々なパラメータを入力することによって視覚的な結果を得ることができる。つまり現象をイメージしやすいという点において、教科書のみから得る知識よりはるかに有効である。その意味において、今回物理シミュレータの作成において用いた技術が、よりよい科学教育を目指す現場の教師にとって多少なりとも手助けになれば幸いである。なお、製作したシミュレータは下記のサイトにおいてインターネットブラウザを介して実際に動作させることが可能である。また、プログラム構造はタイトル、メニュー、シミュレーション実行のそれぞれの画面（図1、①-③）上で、[右クリック] → [ソースの表示] で確認できる。

<http://www.kcat.zaq.ne.jp/aaazo108/sample/> あるいは

<http://natsci.kyokyo-u.ac.jp/~rigaku/forum/4gou/sample>

（『フォーラム理科教育』ホームページ内）

文献

- 西川哲治編 1992. 「改訂版 物理学事典」. 培風館, 東京
高嶋隆一 2000. JAVA を使った物理教育プログラミング. フォーラム理科教育 2 : 41-46.
岡蔵龍一 1999. 「詳解 HTML & JavaScript 辞典」. 秀和システム, 東京
平田邦男 1988. 「新/BASIC による物理」. 共立出版, 東京

『物理現象シミュレータ』作成全般において参考にした文献

- 林晴比古 2001. 「新 Java 言語入門 ビギナー編」. SOFTBANK Publishing, 東京
河西朝雄 2001. 「入門ソフトウェアシリーズ Java 言語」. ナツメ出版, 東京