

教育機関で利用されるウェブフィルタリング ソフトウェアの問題点と改良の研究

卒業論文 II^{*1}

07L503013

白崎智哉^{*2}

2012 年 1 月 12 日

^{*1} 山口毅先生

^{*2} 社会学科 shrskt@gmail.com / 40 字 * 38 行 * 31 ページ, 400 字詰め原稿用紙 118 ページ相当。

目次

第 1 章	序論	1
1.1	なぜこのテーマを設定したか	1
1.2	用語の定義	1
1.3	先行研究	2
第 2 章	問題設定	4
2.1	ウェブフィルタリングソフトウェアには教科書と同じくある種の恣意性があるのではないか	4
2.2	ウェブフィルタリングソフトウェアを複数用意すればよいのではないか	4
2.3	教科書も複数用意すればよいのではないか	5
第 3 章	論証の方策と妥当性	6
3.1	各ウェブフィルタリングソフトウェアの遮断の傾向を見る	6
3.2	ウェブフィルタリングソフトウェアを複数用意する	6
3.3	教科書を複数用意している学校，自治体を調べる	6
第 4 章	論証	7
4.1	各ウェブフィルタリングソフトウェアの遮断の傾向を見た	7
4.2	ウェブフィルタリングソフトウェアを複数用意した	10
4.3	教科書を複数用意している学校，自治体を調べた	12
第 5 章	結論	14
5.1	わかったこと	14
5.2	残った課題	15
	謝辞	16
	参考文献	17
付録 A	ソースコード	21
A.1	ウェブフィルタリングソフトウェア比較ツール	21

A.2	ウェブフィルタリングソフトウェア評価ツール	24
索引		31

第 1 章

序論

1.1 なぜこのテーマを設定したか

重要なメールを，高度な統計処理を応用した迷惑メールフィルタに削除されてしまうように，初等中等教育の現場で「有害情報」を遮断するために利用されるウェブフィルタリングソフトウェアも，誤った判断を下すことがある。この問題は，ウェブフィルタリングソフトウェアを複数用意して，生徒自身に比較させることで，トータルでウェブフィルタリングソフトウェアの信頼性を向上させることで部分的に解決できるものと考えられる。

また，ウェブフィルタリングソフトウェアが政治的に微妙な歴史用語を遮断している，あるいは遮断していないとすれば，歴史教科書とウェブフィルタリングソフトウェアは，構造的に同じ問題を抱えていることになる。ウェブフィルタリングソフトウェアと同じ構造を持つ歴史教科書においても，複数用意して生徒自身に比較させることで，「歴史教科書問題」の一部を回避できるのではないかと考えた。

つまり，この論文の目的は，次の 2 点である。

- ウェブフィルタリングソフトウェアを複数用意すれば，それぞれのウェブフィルタリングソフトウェアの信頼性を比較して検討できることを証明すること。
- ウェブフィルタリングソフトウェアと歴史教科書を比較して，歴史教科書問題に，歴史教科書を複数用意して比較する方法論を持ち込めないか検討すること。あるいは，逆に，歴史教科書の比較研究をウェブフィルタリングソフトウェアに持ち込めないか検討すること。

1.2 用語の定義

1.2.1 ウェブフィルタリングソフトウェア

以降，初等中等学校において，生徒の使用するコンピュータの上で有害なコンテンツの遮断を行うクライアントアプリケーション型，生徒の使用するコンピュータのトラフィックを監視してサーバ上で遮断を行うアプライアンス型，生徒の使用するコンピュータのト

表 1.1 ウェブフィルタの種類

種別	学校での採用率	
クライアントアプリケーション型	併せて 61.7%	個々の端末上で遮断するフィルタ
アプライアンス型		校内計算機室のサーバ上で遮断するフィルタ
プロバイダ型	26.0%	プロバイダ側で遮断するフィルタ

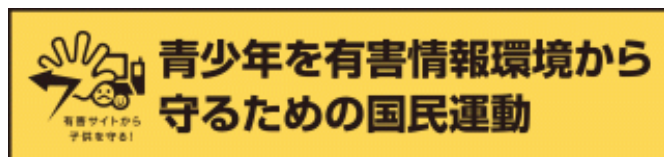


図 1.1 国民運動の広報ロゴマーク

出典：内閣官房 IT 担当室「内閣官房インターネット上の違法・有害情報対策」

ラフィックをプロバイダ側で監視して遮断を行うプロバイダ型のコンテンツフィルタ（表 1.1）をまとめてウェブフィルタリングソフトウェアと呼称する。

また、ウェブフィルタリングソフトウェアは、子供と有害情報の接触を防ぐ装置（図 1.2）としての意味に限定する。本論では、組織の内部統制^{*1}の手段としての側面およびコンピュータシステムを保護する装置としての側面は取り扱わない。

1.2.2 歴史教科書問題

教科書問題は、突出して議論される歴史教科書問題を選択した。歴史教科書以外の教科書問題については、本論では取り扱わない。また、今回は、現行制度内で問題解決を図るため、検定制度には触れない。

1.3 先行研究

さて、ここ数年、犯罪被害を防ぐ目的や、ICT（情報通信技術）教育、メディアリテラシー教育から見たウェブフィルタリングソフトウェアの有効性は、文部科学省、経済産業省、警察庁、通信事業者団体などによって検討されている。国民運動（図 1.1）などのさまざまな議論の具体的な成果は、2009 年 4 月 1 日施行の「青少年が安全に安心してインターネットを利用できる環境の整備等に関する法律」のような法制化に見て取れる。

しかしながら、ウェブフィルタリングソフトウェアの精度や欠陥、子供から見たウェブフィルタリングソフトウェアの問題点、これらの問題を解決するためのプロセスに注目する研究は現在ほとんど行われていない。

^{*1} いわゆる日本版 SOX 法に準拠した情報漏洩の防止など。

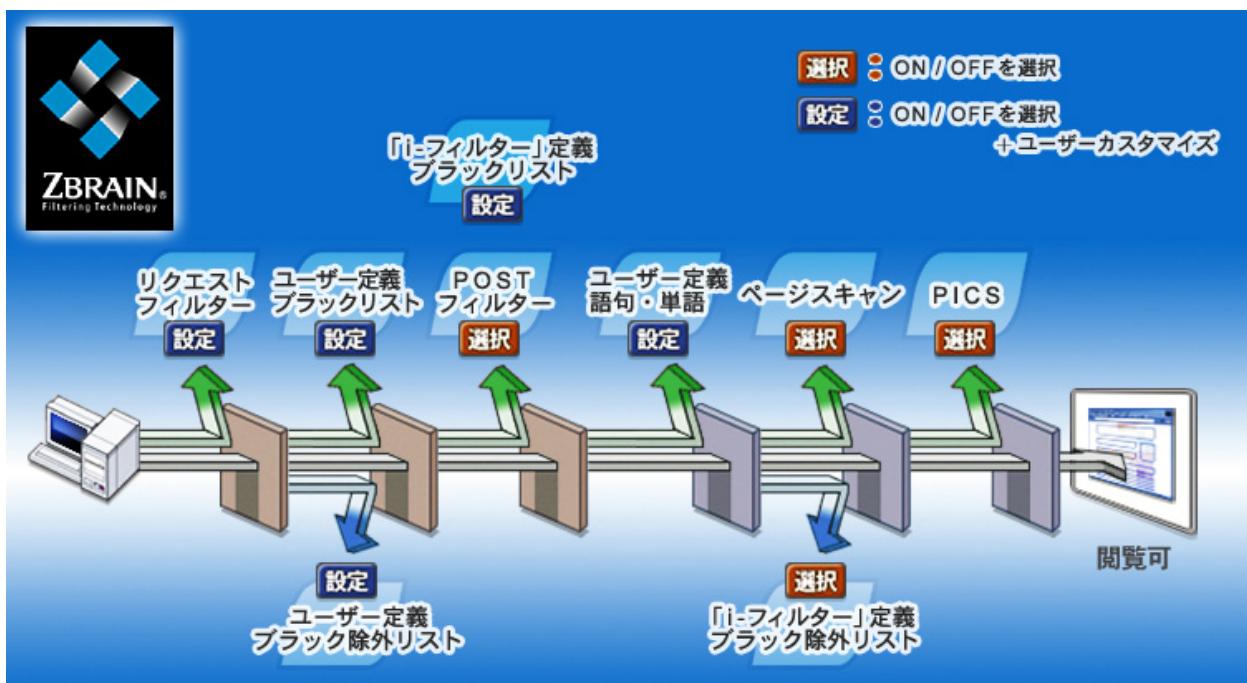


図 1.2 Web フィルタリングの仕組み

出典：デジタルアーツ「「i-フィルター」のテクノロジー」

第 2 章

問題設定

2.1 ウェブフィルタリングソフトウェアには教科書と同じくある種の恣意性があるのではないか

文部科学省の「学校における教育の情報化の実態等に関する調査結果（平成 19 年度）」によれば，わが国の初等中等学校 37,050 校中，37,037 校はインターネットに接続しており，割合では 99.96% になる。また，インターネット接続になんらかのフィルタリングを施している学校の割合は，94.6% である。

このように広く普及している，すべてのウェブフィルタリングソフトウェアの有害情報の遮断の判断は，理想の上では，必ず同一になる。しかし，実際には，製品ごとにそれぞれ別の設計で，別のポリシーを採用している上，学校ごとに個別の禁止ワードを加える以上，どのウェブフィルタリングソフトウェアも，必ずしも同じ有害情報を常に遮断するとは言えない。歴史教科書のポリシーが問題視されるのであれば，ウェブフィルタリングソフトウェアのポリシーも同じように問題視されておかしくない。さらに，禁止ワードは，その特性上，外に開示されないため，管理者は，禁止ワードを書き加えて，正規の情報を遮断するような恣意的な運用もできる。

2.2 ウェブフィルタリングソフトウェアを複数用意すればよいのではないか

それぞれのウェブフィルタリングソフトウェアの挙動が違う場合，挙動の違いの証拠を生徒が提出できれば，あるウェブフィルタリングソフトウェアの挙動の不審を学校側の管理者に説明できる。また，あるウェブフィルタリングソフトウェアに問題のないことも同じように説明できる。あるいは，全く問題ないと考えられるページを，すべてのウェブフィルタリングソフトウェアが遮断している場合にも証拠を用意できる。

表 2.1 歴史教科書とウェブフィルタリングソフトウェアの比較

	歴史教科書	ウェブフィルタリングソフトウェア
検定制度		
複数採用		(全体の 6.9%)
受験との関わり	強い	弱い

2.3 教科書も複数用意すればよいのではないか

小森陽一らの『歴史教科書何が問題か 徹底検証 Q&A』では、程度を除けば、すべての歴史教科書になんらかの問題があるとしている。無謬の教科書を用意できないのであれば、すべての年代のすべての教科書を揃えて、事項を比較できればよいのではないか。もしくは、現行のすべての歴史教科書の記述の差異のみを集めた、注釈書のようなものを常に発行すれば、より簡単に比較できるのではないか。

比較の手間、膨大な教科書の輸送や、保管の容積の問題は、電子教科書からもアプローチできる。

第3章

論証の方策と妥当性

3.1 各ウェブフィルタリングソフトウェアの遮断の傾向を見る

複数のウェブフィルタリングソフトウェアの遮断の傾向が全く同一であるのか、それぞれ違うのかを調べる。ウェブフィルタリングソフトウェアの特性が異なることを証明することは、精度の悪いウェブフィルタリングソフトウェアの存在を立証することに繋がる。よってこの文脈で、各ウェブフィルタリングソフトウェアの遮断の傾向を見ることは妥当であると考えられる。

3.2 ウェブフィルタリングソフトウェアを複数用意する

ウェブフィルタリングソフトウェアを複数用意できれば、遮断の傾向を比較できる。ウェブフィルタリングソフトウェア同士の比較が可能になれば、ウェブフィルタリングソフトウェアの精度や不具合を利用者自身が確認できるようになると考えられる。よって、ウェブフィルタリングソフトウェアを複数用意することは妥当であると考えられる。

3.3 教科書を複数用意している学校、自治体を調べる

歴史教科書を複数用意している学校、自治体を調べることで、ウェブフィルタリングソフトウェアに既存の方法論を応用できる可能性がある。よって、教科書を複数用意している学校、自治体を調べることは、この文脈では妥当であると考えられる。

第 4 章

論証

4.1 各ウェブフィルタリングソフトウェアの遮断の傾向を見た

ウェブフィルタリングソフトウェアの遮断の傾向を確認する方法はいくつか考えられる。

- 実際に教育用コンピュータからウェブフィルタリングソフトウェアを経由してインターネットに接続している中高生に質問紙調査ないし聞き取り調査を行う
- 無作為な URL を用意して遮断されるかどうかをウェブフィルタリングソフトウェアごとに記録する

1 番目の調査方法は、利用者の記憶に頼った信頼性の低いデータになると考えられる。

2 番目の方法は、予め用意する URL 以外は高い客観性を持つ点と、複数のウェブフィルタリングソフトウェアを同じ条件で確認できる点で、1 番目の方法より優れる。

そこで、2 番目の方法を発展させて、ウェブ上でいくつかの歴史用語を日本語版 Google で検索して、検索結果の 1 ページ目に含まれる各検索結果がそれぞれ遮断されているかどうか調べるツールを作成した。

4.1.1 ウェブフィルタリングソフトウェア評価ツール

このツールでは、始めに、全国歴史教育研究協議会編『日本史 B 用語集改訂新版』の「近代日本と世界大戦」章「第二次世界大戦」節のすべての単語 168 単語を使用して、単語ごとに Google 検索を行い、平均 10 ページ分の検索結果の URL を収集する（図 4.1）。次に、収集した URL に順次アクセスを試みる（図 4.2）。通常、ウェブフィルタリングソフトウェアにアクセスを遮断された場合は、ウェブフィルタリングソフトウェア特有の URL に遷移するか、ブラウザのタイトルバーが書き換わるため、アクセス遮断の有無を URL ごとに自動で記録できる。

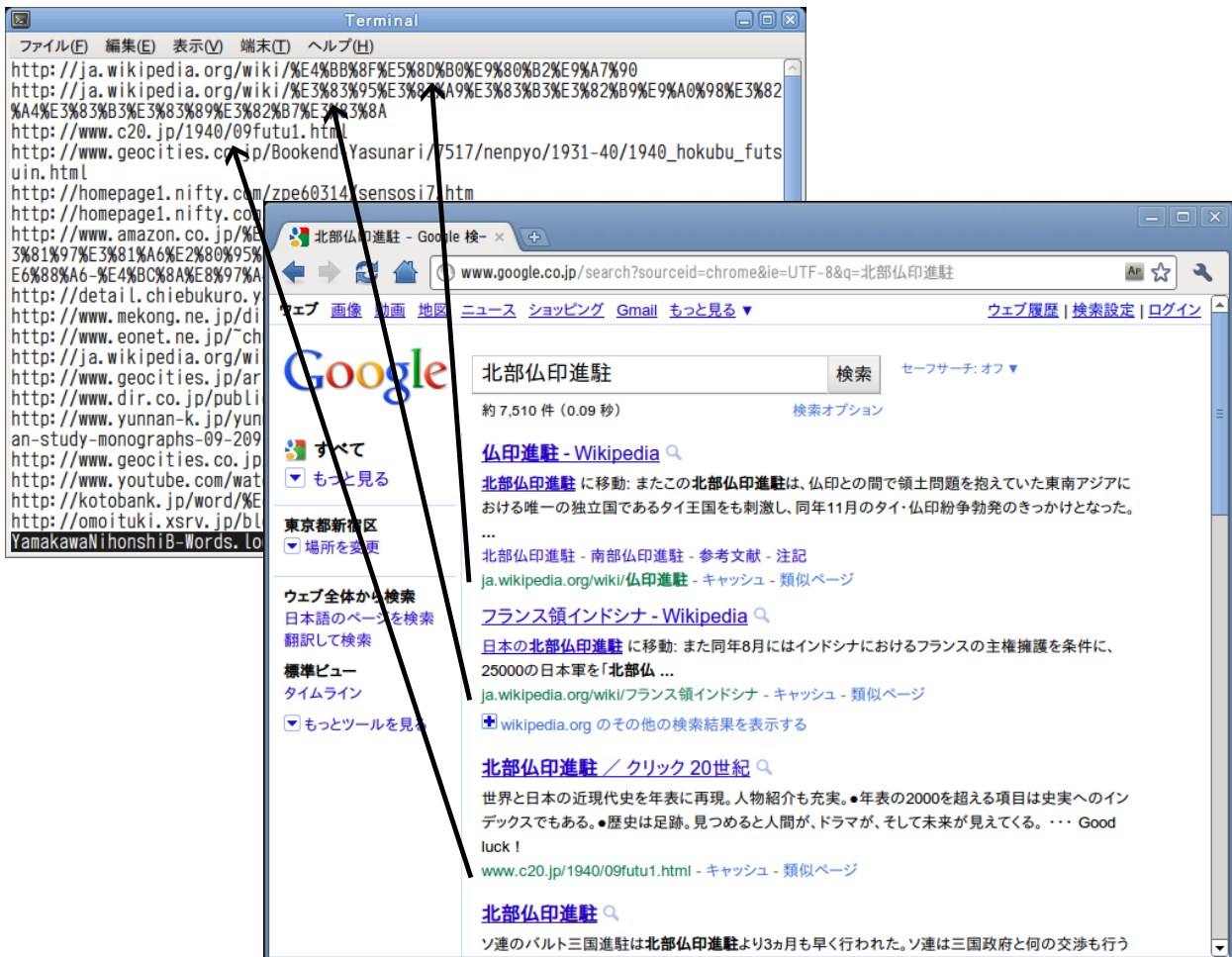


図 4.1 拙作 YamakawaNihonshiB-UrlCollector.rb 動作イメージ

ウェブフィルタリングソフトウェアなしの場合

まず、それぞれのウェブフィルタリングソフトウェアに通すための URL を、ウェブフィルタリングソフトウェアなしで収集した結果、1,769 ページの URL を収集できた。そのうち 13 ページにアクセスできなかった^{*1}。また、PDF 文書は 18 文書あった。

i-フィルターの場合

1 番目のウェブフィルタリングソフトウェアとして、Digital Arts 社製 i-フィルター 6.0 を用意した。i-フィルターの設定は、インストール時のデフォルトを使用した。テストする URL は、先に取得した 1,769 ページから、アクセスできない 13 ページと、PDF 文書の 18 文書を引いた 1,738 ページ分の URL を使用した。

^{*1} エラーの内訳を次に示す。アクセス拒否：2 ページ、未検出：2 ページ、サービス利用不可：3 ページ、タイムアウトエラー：7 ページ。

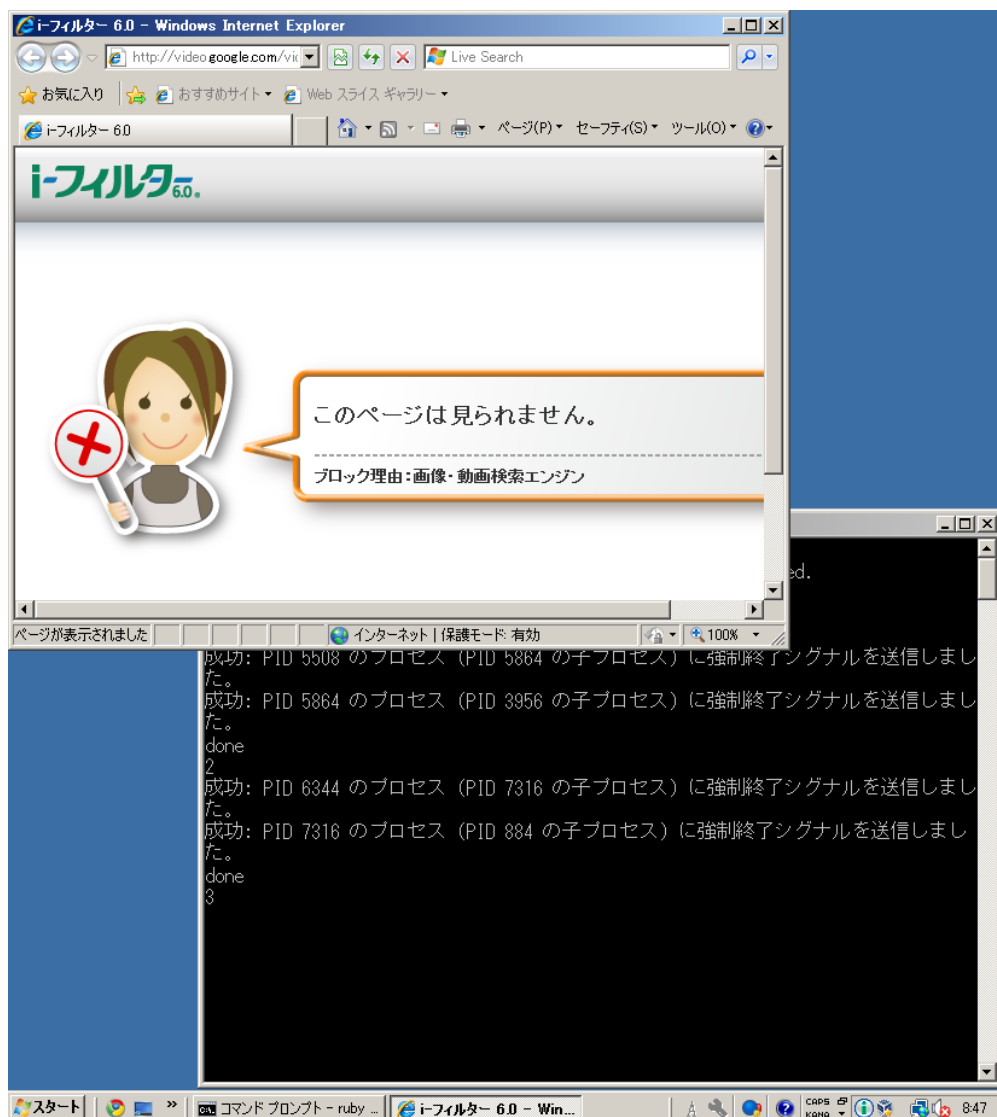


図 4.2 拙作 ieLauncher.rb/LastTitleRecorder 実行画面のハードコピー

結果, 5 ページへのアクセスを i-フィルターに遮断された。

Windows Live ファミリーセーフティの場合

2 番目のウェブフィルタリングソフトウェアとして, Microsoft 社製の Windows Live ファミリーセーフティ (2011) を用意した。Windows Live ファミリーセーフティの設定は, インストール時のデフォルトを使用した。テストする URL は, 先に取得した 1,769 ページから, アクセスできない 13 ページと, PDF 文書の 18 文書を引いた 1,738 ページ分の URL を使用した。

結果, 96 ページへのアクセスを Windows Live ファミリーセーフティに遮断された。

4.2 ウェブフィルタリングソフトウェアを複数用意した

次に、ウェブフィルタリングソフトウェア同士の違いを比較するために、ウェブフィルタリングソフトウェアを複数用意したい。ウェブフィルタリングソフトウェアを複数用意して教育現場で運用する方法はいくつか考えられる。

- 1 台だけ別のフィルタをインストールしたコンピュータを用意する方法
- フィルタの先にフィルタを置いて、フィルタを重ねる方法
- コンピュータ側に複数のフィルタに問い合わせるプログラムを用意する方法

1 番目の方法は、一番安価である。デメリットとしては、利用者に物理的な移動を強い点、および管理者に不当な遮断を訴える際の証拠提出が困難である点を挙げられる。

2 番目の方法は、一番フェイルセーフと言える。デメリットとしては、手前に目の細かいフィルタを置くとどの段階のウェブフィルタリングソフトウェアに遮断されたのかわかりにくい点、さらに一つ目より、不当な遮断を訴えにくい点を挙げられる。この方法は、実際に採用例があり、インターネットに接続している初等中等学校の 6.9% は、この方法を採用している。

3 番目の方法は、それ以外の方法と比べて、比較の容易さと、遮断の不整合の申請し易さにおいて優位である。アプライアンス型のウェブフィルタリングソフトウェアを 2 台以上必要とするため、構成によってはかなり高価になるデメリットがある。クライアントアプリケーション型とアプライアンス型の構成では、二つ目の方法と同程度か、やや上回る程度のコストになる。

そこで、3 番目の方法の試作として、複数のウェブフィルタリングソフトウェアを画面上で比較できるツール（図 4.3，4.5）を作成した。

4.2.1 ウェブフィルタリングソフトウェア比較ツール

このプログラム起動して「取得」をクリックすると、プログラムは、起動中のウェブブラウザから現在の URL を取得して、複数のウェブフィルタリングソフトウェアに問い合わせを行う。各ウェブフィルタリングソフトウェアごとの遮断の有無を並べて表示することで、利用者はウェブフィルタリングソフトウェア同士の不整合を簡単に比較検討できる。

ここで用意する比較用のウェブフィルタリングソフトウェアは、プログラムの都合上すべてアプライアンス型である必要がある。また、ここで 3 種類を想定している比較用のウェブフィルタリングソフトウェアは、予算の許す限り増設できる。

このプログラムでは、結果を管理者にメールで送付する手段を用意して、学校 - 生徒間の証拠ベースの合意形成の可能性を高めた。

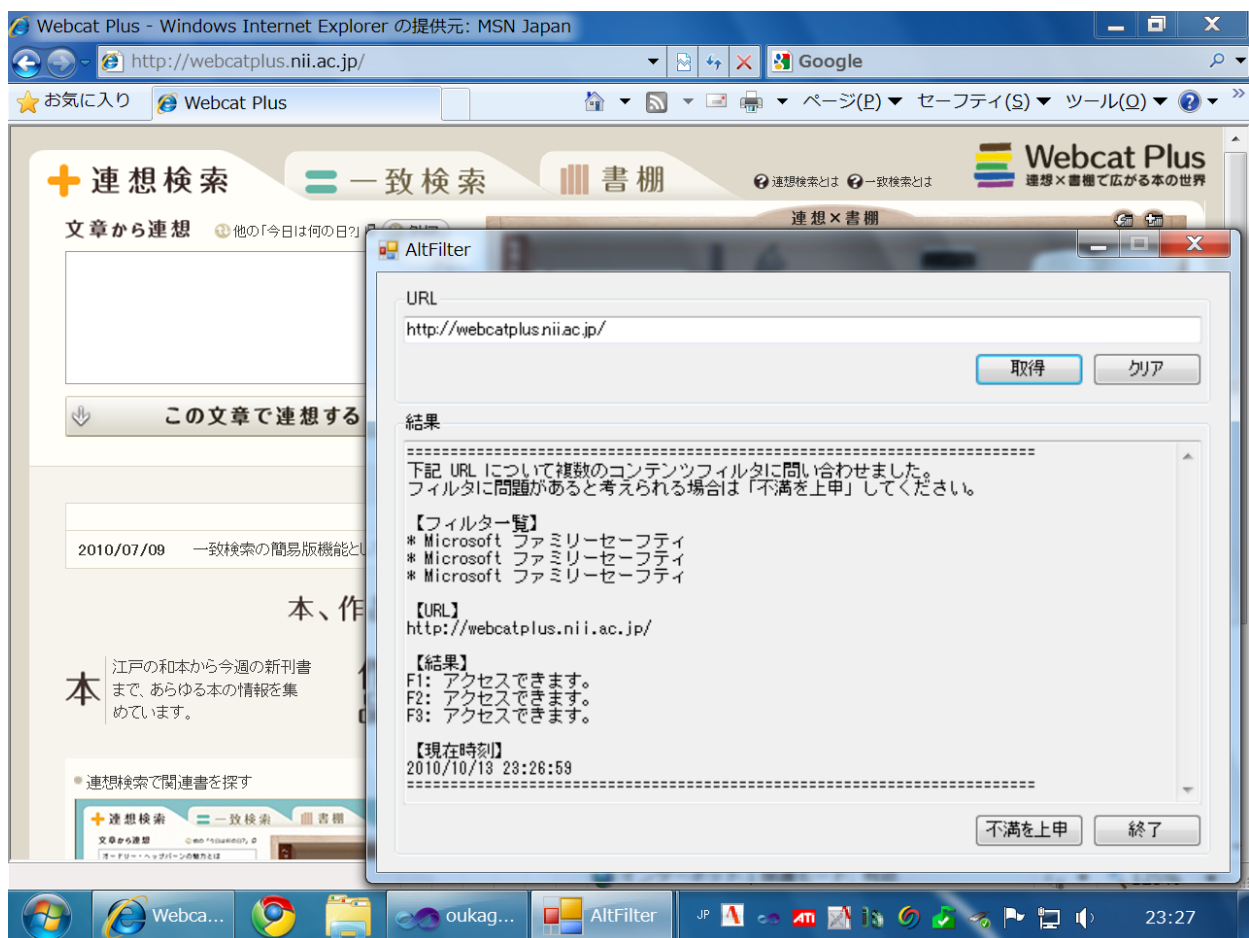


図 4.3 拙作 AltFilter 実行結果

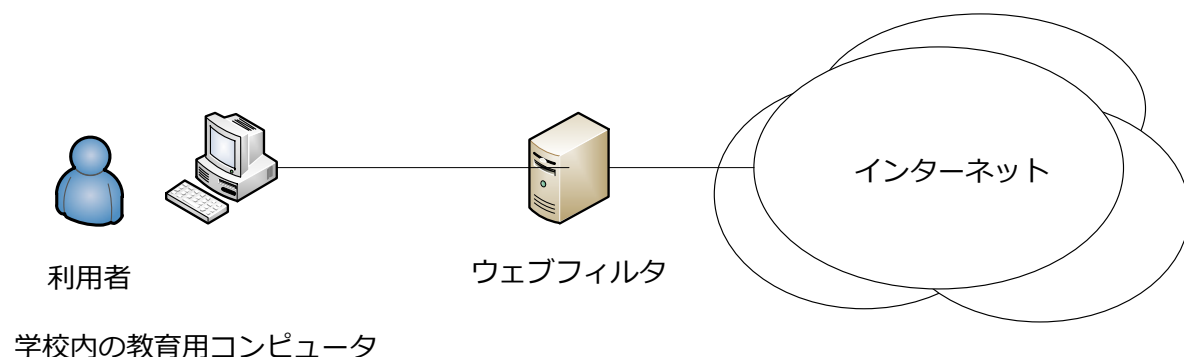


図 4.4 通常のネットワーク構成の概念図

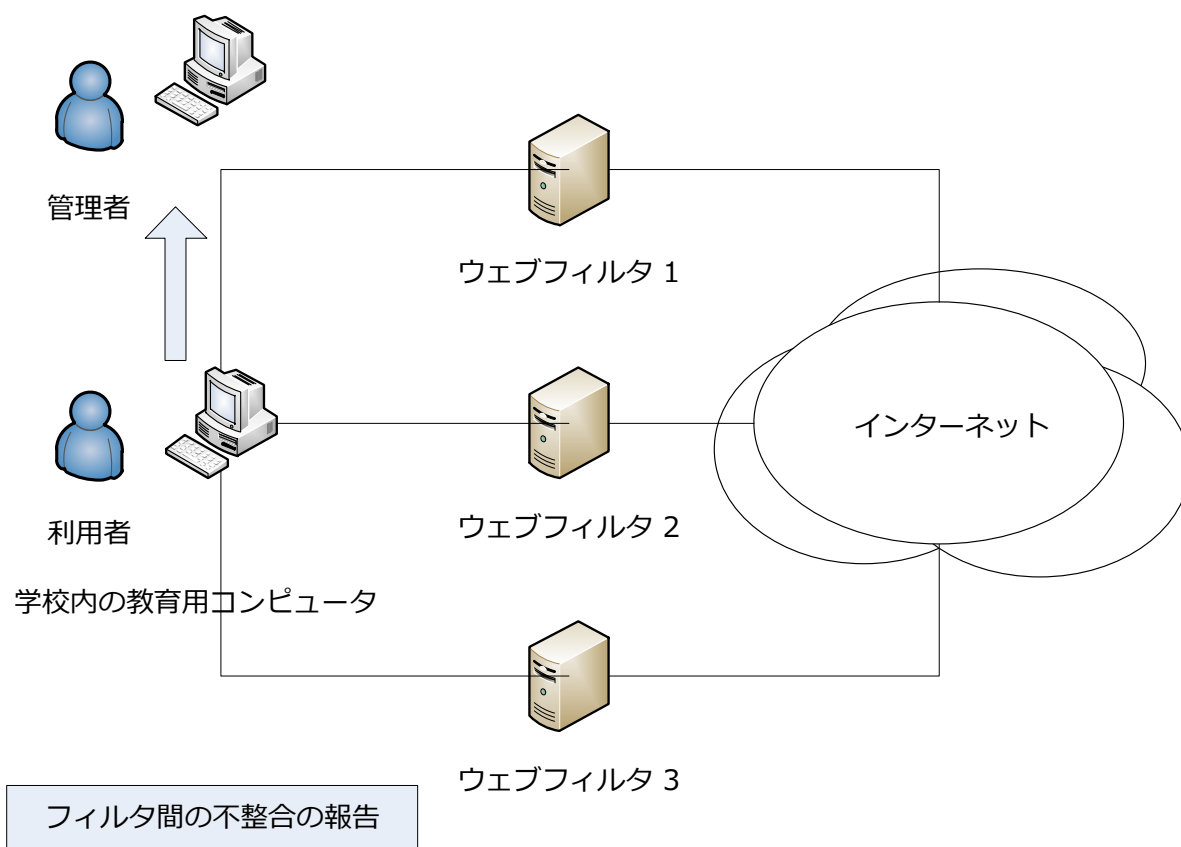


図 4.5 AltFilter 概念図

4.3 教科書を複数用意している学校，自治体を調べた

財団法人教科書研究センター附属教科書図書館や，国立教育政策研究所教育研究情報センター教育図書館などの特殊な図書館を除いては，検定済みの教科書は，各地の大学図書館のほか，各自治体の一部の図書館で閲覧のサービスを確認できた。

また，全国 880 箇所自治体図書館や公立学校，市役所などに設置される教科書センター（図 4.6）においても検定済み教科書を常時閲覧できる。

検定済み教科書の貸し出しサービスは，自治体の図書館では行われていないことが多い。

4.3.1 教科書センター

教科書センターとは，1956 年以来，各都道府県が，教育者や住民向けに設置している常設の教科書展示場である。2010 年 5 月現在，全国に 880 箇所設置されている。

教科書センターでは，小学校用教科書，中学校用教科書，高等学校用教科書，特別支援

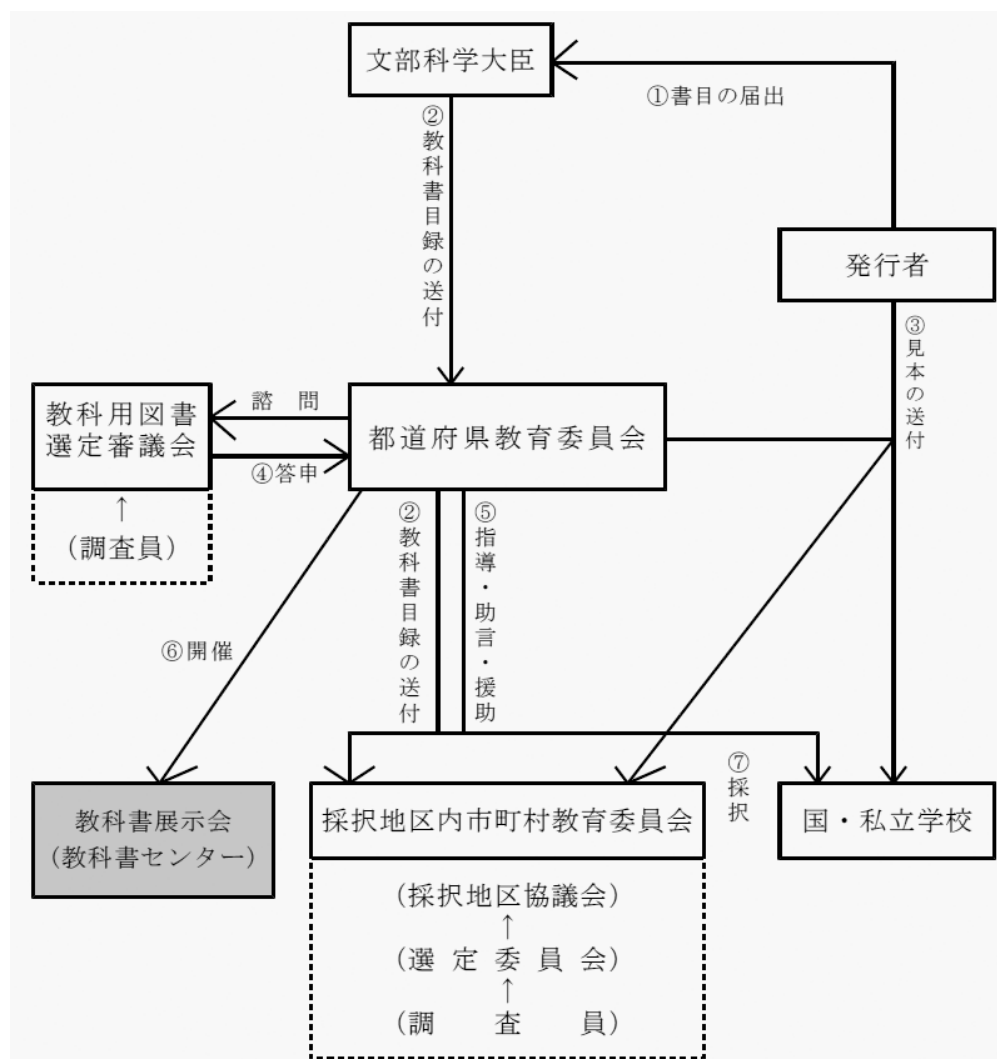


図 4.6 義務教育諸学校用教科書の採択の仕組み

出典：文部科学省「教科書制度の概要」より一部改変

学校用教科書、一般図書*²（特別支援学校用・学級用）のほか、現行でない古い教科書も閲覧できる。ただし、貸し出し、複製のサービスは行われていない。

^{*2} 学校教育法附則第 9 条により，教科書として利用可能な図書を指す。

第 5 章

結論

5.1 わかったこと

繰り返すと，ウェブフィルタリングソフトウェアは利用者の比較によって，その精度をモニタリングできる。さらに，今回作成した AltFilter では，モニタリングの先に，ウェブフィルタリングソフトウェアそのものと，その設定についての管理者との合意形成プロセスがある。

また，歴史教科書の比較については，全国の教科書センターで比較できることがわかった。

5.1.1 各ウェブフィルタリングソフトウェアの遮断の傾向

各ウェブフィルタリングソフトウェアの遮断の傾向は，山川出版社の発行する『日本史 B 用語集改訂新版』の 168 単語を元に，Google の検索結果から収集した 1,738 の URL では，i-フィルタで 5 ページの遮断，Windows Live ファミリーセーフティで，96 ページの遮断があり，ウェブフィルタリングソフトウェアごとに遮断の傾向が異なることを実証できた。

この結果から，程度や設定の差はあれ，精度のよくないウェブフィルタリングソフトウェアは存在するものと推測できる。

5.1.2 ウェブフィルタリングソフトウェアの比較

ウェブフィルタリングソフトウェアの比較の方法については，クライアントコンピュータ用のウェブフィルタリングソフトウェア比較ツールを作成することで，利用者の手元で複数のウェブフィルタリングソフトウェアを比較検討できることを実証できた。

この結果から，ウェブフィルタリングソフトウェアの複数採用は，生徒側から学校側への，ウェブフィルタリングソフトウェアの不具合の証拠提出に有効であるといえる。

5.1.3 教科書を複数用意している学校，自治体

教科書を複数用意している自治体は，それほど多くなかった。例えば，横浜市立図書館は，教科書センター指定を受ける以前の，1994 年度から 2006 年度は採択教科書のみのも蔵であるため，1994 年度から 2006 年度の横浜市立図書館を利用する初等中等学校の生徒は教科書同士を比較するような読み方はできない。

加えて，教科書センターの教科書は，基本的に検定済みの教科書を展示しているため，検定済みを通らない教科書と検定済み済み教科書の教科書を比較するような読み方もできない。

5.2 残った課題

次の 3 点は課題として残った。

- ウェブフィルタリングソフトウェアは受験の点数に影響するかどうか不明な点
- 歴史教科書問題の根本的な解決ではない点
- 歴史教科書の比較のコストが多大である点

5.2.1 ウェブフィルタリングソフトウェアは受験の点数に影響するかどうか不明な点

教科書問題は，受験への影響も問題視されている。同じようにウェブフィルタリングソフトウェアも受験に影響すると仮定して，点数で問題を立証できるとしても，測定が難しい。

5.2.2 歴史教科書問題の根本的な解決ではない点

歴史教科書問題の観点からは，歴史教科書問題の根本的な解決でなく，回避である点で，教科書の比較という手法に限界があると考えられる。

また，比較にかかる時間的不経済を生徒に押し出している点で問題が残る。

5.2.3 歴史教科書の比較のコストが多大である点

コスト意識の観点では，教育現場で，中高生に実際に教科書を比較してもらえるかどうか疑問が残る。導入の賛否はともあれ，電子教科書であれば，現在の教科書よりはるかに，本論のウェブフィルタリングソフトウェアのように比較しやすくなるものと考えられる。

謝辞

本論は，山口毅先生にご指導を賜りました。心より御礼申し上げます。

一部は初年次の導入教育から高城和義先生にご指導を賜りました。厚く御礼申し上げます。

プログラムの作成に使用した Microsoft Visual Studio 2010 Professional は，マイクロソフト株式会社の学生支援プログラムである Microsoft DreamSpark プログラムを経由して入手したものです。また，基本的な Ruby 言語環境のほか，mechanize を使用しました。それぞれのソフトウェアの世界中の開発者の方々に御礼申し上げます。

参考文献

- [1] Ayres, Ian, 2007, *Super crunchers: why thinking-by-numbers is the new way to be smart*, New York: Bantam Books. (= 2010 , 山形浩生訳 , 『その数学が戦略を決める』文藝春秋 .)
- [2] Fromm, Erich, 1941, *Escape from freedom*, New York: Farrar & Rinehart, inc. (= 1951 , 日高六郎訳 , 『自由からの逃走』東京創元社 .)
- [3] Lessig, Lawrence, 2006, *Code*, New York: Basic Books.(= 2007 , 山形浩生訳 , 『Code』翔泳社 .)
- [4] Olsen, Stefanie, 2006, "Kids outsmart Web filters," CNET News.com, (Retrieved August 10, 2011, http://news.cnet.com/2009-1041_3-6062548.html). (= 2006 , 坂和敏訳 , 「アクセス規制をすり抜ける子供たち 米の教育現場から」, CNET Japan ウェブサイト , (2011 年 8 月 10 日取得 , <http://japan.cnet.com/news/commentary/20102487/>).)
- [5] Orwell, George, 2003, *Nineteen Eighty-Four*, London: Penguin. (= 2009 , 高橋和久訳 , 『一九八四年 [新訳版]』早川書房 .)
- [6] Reichman, Henry, 2001, *Censorship and selection: issues and answers for schools*, Chicago: American Library Association. (= 2002 , 川崎佳代子・川崎良孝訳 , 『学校図書館の検閲と選択 アメリカにおける事例と解決方法 (第 3 版)』京都大学図書館情報学研究会 .)
- [7] Sawyer, Keith ed., 2006, *The Cambridge Handbook of the Learning Sciences*, Cambridge: Cambridge University Press. (= 2009 , 森敏昭・秋田喜代美監訳 , 『学習科学ハンドブック』培風館 .)
- [8] 朝日新聞社 , 2011a , 「携帯電話のフィルタリング義務 県が罰則付き条例案 来年 1 月の施行めざす」『朝日新聞』2011 年 3 月 3 日群馬朝刊: 35 , (2011 年 5 月 12 日取得 , 聞蔵 II ビジュアル) .
- [9] , 2011b , 「どうする? 親の出番 (2) 「危ないこと」を体験 多少のケガ覚悟 見守って」『朝日新聞』2011 年 9 月 11 日東京朝刊: 29 , (2011 年 9 月 14 日取得 , 聞蔵 II ビジュアル) .
- [10] , 2011c , 「非出会い系で性被害の少女増加 携帯の”制限”不徹底」『朝日新聞』2011 年 2 月 19 日宮城朝刊: 31 , (2011 年 5 月 12 日取得 , 聞蔵 II ビジュアル) .

- [11] 東浩紀, 2007, 『情報環境論集 東浩紀コレクション S』講談社.
- [12] 東浩紀・大澤真幸, 2003, 『自由を考える 9・11以降の現代思想』日本放送出版協会.
- [13] アルプスシステムインテグレーション, 2011, 「福島県教育センター導入事例」, ALSI, (2011年5月16日取得, http://www.alsi.co.jp/casestudy/fukushima_edu.html).
- [14] 家永三郎・暉峻淑子述, 1994, 『なぜ教科書裁判をたたかったのか』岩波書店.
- [15] 稲葉振一郎, 2008, 『「公共性」論』NTT出版.
- [16] 岡村久道, 2008, 「インターネット関連法律の全体動向」日本インターネット協会編『インターネット白書 2008』インプレス.
- [17] 子どもたちのインターネット利用について考える研究会, 2009, 「中高生のお子さんを持つ保護者のためのインターネットセーフティガイド」, 子どもたちのインターネット利用について考える研究会ウェブサイト, (2011年9月14日取得, (2011年5月16日取得, http://www.child-safenet.jp/material/index_02.html).
- [18] 小森陽一・安丸良夫・坂本義和編, 2001, 『歴史教科書何が問題か 徹底検証 Q&A』岩波書店.
- [19] 小山常実, 2001, 『歴史教科書の歴史』草思社.
- [20] 佐々木俊尚, 2008, 「青少年ネット規制法案をめぐる論点」日本インターネット協会編『インターネット白書 2008』インプレス.
- [21] 社会科教科書執筆者懇談会編, 1984, 『教科書問題とは何か』未来社.
- [22] 自由国民社, 2010, 「フィルタリング」『現代用語の基礎知識』, (2011年5月12日取得, 現代用語の基礎知識).
- [23] 全国歴史教育研究協議会編, 2000, 『日本史 B 用語集 改訂新版』山川出版社.
- [24] 杉原里美, 2011a, 「どうする? 青春期のネット利用 1 情報を選びとる力, 養う」『朝日新聞』2011年6月12日東京朝刊: 29, (2011年9月14日取得, 聞蔵 II ビジュアル).
- [25] , 2011b, 「どうする? 青春期のネット利用 2 判断力を見極め段階的に」『朝日新聞』2011年6月19日東京朝刊: 29, (2011年9月14日取得, 聞蔵 II ビジュアル).
- [26] , 2011c, 「どうする? 青春期のネット利用 3 アクセス制限, レベル様々」『朝日新聞』2011年6月26日東京朝刊: 23, (2011年9月14日取得, 聞蔵 II ビジュアル).
- [27] , 2011d, 「どうする? 青春期のネット利用 4 加害者になる危険も確認」『朝日新聞』2011年7月3日東京朝刊: 31, (2011年9月14日取得, 聞蔵 II ビジュアル).
- [28] 杉原里美・前田育穂, 2011, 「どうする? 青春期のネット利用 5 日頃から親子で話題に」『朝日新聞』2011年7月10日東京朝刊: 29, (2011年9月14日取得, 聞蔵 II ビジュアル).

- [29] 総務省, 2008, 「青少年が安全に安心してインターネットを利用できる環境の整備等に関する法律」, 法令データ提供システム, (2010 年 12 月 1 日取得, <http://law.e-gov.go.jp/announce/H20H0079.html>).
- [30] 丹治吉順, 2011, 「ニュースがわからん! 中東のデモ, ネット遮断って何?」『朝日新聞』2011 年 2 月 25 日東京朝刊: 2, (2011 年 5 月 12 日取得, 聞蔵 II ビジュアル).
- [31] デジタルアーツ, 2011, 「i-フィルター」のテクノロジー, デジタルアーツ社ウェブサイト, (2011 年 6 月 10 日取得, http://www.daj.jp/filter/ifl_how.htm).
- [32] , 2011b, 「特許技術と独自のノウハウ」, デジタルアーツ社ウェブサイト, (2011 年 6 月 10 日取得, <http://www.daj.jp/cs/ifilter/trust/>).
- [33] 内閣官房 IT 担当室, 2007, 「青少年を有害情報環境から守るための国民運動」, 内閣官房インターネット上の違法・有害情報対策ウェブサイト, (2010 年 12 月 1 日取得, <http://www.it-anshin.go.jp/activity/>).
- [34] 内閣府・総務省・経済産業省, 2008, 「青少年が安全に安心してインターネットを利用できる環境の整備等に関する法律施行令(案)」に対する意見募集の結果について」, 内閣府内閣府共生社会政策担当ウェブサイト, (2011 年 8 月 10 日取得, <http://www8.cao.go.jp/youth/iken/internet/kekka.html>).
- [35] , 2009, 「青少年インターネット環境整備関係法令条文解説」, 内閣府内閣府共生社会政策担当ウェブサイト, (2011 年 8 月 10 日取得, <http://www8.cao.go.jp/youth/youth-harm/law/pdf/kaisetsu.pdf>).
- [36] 内閣府共生社会政策統括官, 2011a, 「アメリカ・ドイツにおける青少年のインターネット環境整備状況等調査報告書」, 内閣府内閣府共生社会政策担当ウェブサイト, (2011 年 8 月 10 日取得, http://www8.cao.go.jp/youth/youth-harm/chousa/h22/net-us_de/index.html).
- [37] , 2011b, 「青少年のインターネット利用環境実態調査」, 内閣府内閣府共生社会政策担当ウェブサイト, (2011 年 8 月 10 日取得, <http://www8.cao.go.jp/youth/youth-harm/chousa/h22/net-jittai/pdf-index.html>).
- [38] 新潟市立総合教育センター, 「新潟教科書センター」, 新潟市立総合教育センターウェブサイト, (2010 年 12 月 1 日取得, http://www.netin.niigata.niigata.jp/kyokasho_center/kyokasho_center.html).
- [39] 日本インターネット協会編, 2006, 『インターネット白書 2006』インプレス.
- [40] , 2007, 『インターネット白書 2007』インプレス.
- [41] , 2008, 『インターネット白書 2008』インプレス.
- [42] , 2009, 『インターネット白書 2009』インプレス.
- [43] 松浦正浩, 2010, 『実践! 交渉学 いかに合意形成を図るか』筑摩書房.
- [44] 日本図書館協会, 2001, 「青少年社会環境対策基本法案についての見解」, 日本図書館協会ウェブサイト, (2011 年 8 月 10 日取得, <http://www.jla.or.jp/portals/0/html/kenkai/seisyonen.html>).
- [45] 森亮二, 2009, 「違法有害情報とその規制の現状」日本インターネット協会編『イン

ターネット白書 2009』インプレス。

- [46] 文部科学省初等中等教育局教科書課，2010，「教科書制度の概要」，文部科学省ウェブサイト，(2010年12月1日取得，http://www.mext.go.jp/a_menu/shotou/kyoukasho/gaiyou/04060901.htm)。
- [47] 文部科学省初等中等教育局参事官付情報網整備室，2008，「学校における教育の情報化の実態等に関する調査結果(平成19年度)」，文部科学省ウェブサイト (2010年12月1日取得，http://www.mext.go.jp/a_menu/shotou/zyouhou/08092209.htm)。
- [48] 文部省初等中等教育局長，1990，「教科書採択の在り方の改善について(通知)」，文部科学省ウェブサイト，(2010年12月1日取得，http://www.mext.go.jp/a_menu/shotou/kyoukasho/kaizen.htm)。
- [49] 横浜市教育委員会事務局中央図書館調査資料課，2009，「教科書を探す」，横浜市立図書館ウェブサイト，(2010年12月1日取得，<http://www.city.yokohama.jp/me/kyoiku/library/theme/kyoukasyo.html>)。
- [50] 読売新聞，2011，「ネット接続9社 児童ポルノ強制遮断開始 契約世帯の6割」『読売新聞』2011年4月21日東京夕刊: 11，(2011年5月12日取得，ヨミダス歴史館)。

付録 A

ソースコード

A.1 ウェブフィルタリングソフトウェア比較ツール

A.1.1 AltFilter

次のプログラムの作成には、Microsoft Visual Basic 2010 を使用した。また、動作環境は、Microsoft Windows XP Service Pack 3 以降の OS、Microsoft Internet Explorer 7.0 以降のウェブブラウザ、Microsoft .NET Framework 3.5 Service Pack 1 以降の .NET 実行環境を想定した。

例示用のウェブフィルタリングソフトウェアは、すべて Microsoft Windows Live ファミリーセーフティを使用した。

ソースコード A.1 AltFilter (Form1.vb)

```

1 Public Class Form1
2
3     Private Sub Button2.Click(ByVal sender As System.Object, ByVal e As System.
        EventArgs) Handles Button2.Click
4         ' テキストボックス内の URL のクリア
5         TextBox1.Text = ""
6     End Sub
7
8     Private Sub Button1.Click(ByVal sender As System.Object, ByVal e As System.
        EventArgs) Handles Button1.Click
9         Dim urlList As New ArrayList()
10        Dim shw = New SHDocVw.ShellWindows
11        Dim ie As SHDocVw.ShellBrowserWindow = Nothing
12        Dim count As Integer
13        Dim urlVar As String
14
15        ' このツールのほかに起動中の Microsoft Internet Explorer から URL を取得する。
16        For Each ie In shw
17            If TypeName(ie.Document) = "HTMLDocumentClass" Then
18                urlList.Add(ie.LocationURL)
19            End If
20        Next ie
21

```



```
22      ' URL を表示する。URL を取得できない場合はエラーメッセージを表示する。
23      count = urlList.Count
24
25      If count = 0 Then
26          ' ステータス表示
27          TextBox2.Text = "URL_の取得に失敗しました。"
28      Else
29          ' 一番上のブラウザの URL の抽出
30          urlVar = urlList(0)
31          ' 一番上のブラウザの URL の表示
32          TextBox1.Text = urlList(0)
33          ' ステータス表示
34          TextBox2.Text = "URL_の取得に成功しました。"
35      End If
36
37      ' 内蔵の不可視 webbrowser で当該 URL をフィルタに通す。
38      WebBrowser1.Navigate(TextBox1.Text)
39
40      End Sub
41
42      Private Sub Button4.Click(ByVal sender As System.Object, ByVal e As System.
43          EventArgs) Handles Button4.Click
44      End Sub
45
46      Private Sub Button3.Click(ByVal sender As System.Object, ByVal e As System.
47          EventArgs) Handles Button3.Click
48          ' 管理者にメールを送信する。
49          Dim message As New System.Net.Mail.MailMessage()
50          message.Subject = "From_AltFilter"
51          ' メール本文に結果を挿入する。
52          message.Body = TextBox2.Text
53          ' 管理者のメールアドレスを予め入力しておく。
54          Dim client As New System.Net.Mail.SmtpClient("root@example.net")
55          client.Send(message)
56      End Sub
57
58      Private Sub WebBrowser1.DocumentCompleted(ByVal sender As System.Object, ByVal
59          e As System.Windows.Forms.WebBrowserDocumentCompletedEventArgs) Handles
60          WebBrowser1.DocumentCompleted
61          Dim F1Var As Boolean
62          Dim F2Var As Boolean
63          Dim F3Var As Boolean
64          Dim F1HumanReadable As String
65          Dim F2HumanReadable As String
66          Dim F3HumanReadable As String
67          Dim getUrl As String
68          Dim result As Byte
69          ' 日付を取得
70          Dim dtNow As DateTime = DateTime.Now
71          ' 変数の初期化
72          F1HumanReadable = "アクセスできます。"
73          F2HumanReadable = "アクセスできます。"
74          F3HumanReadable = "アクセスできます。"
```

```
72
73     ' 戻り URL にフィルタ固有の文字列が含まれている場合に、遮断されたと判断する。
       ( e.g. http://fss.live.com/krl/block )
74     getUrl = WebBrowser1.Url.ToString
75     result = InStr(getUrl, "http://fss.live.com/krl/block")
76     If result > 0 Then
77         F1Var = True
78         F2Var = True
79         F3Var = True
80     End If
81
82     ' 結果に応じてメッセージを書き換える。
83     If F1Var = True Then
84         F1HumanReadable = "遮断されています。"
85     End If
86     If F2Var = True Then
87         F2HumanReadable = "遮断されています。"
88     End If
89     If F3Var = True Then
90         F3HumanReadable = "遮断されています。"
91     End If
92
93     ' 結果の表示
94     TextBox2.Text = "
       =====
       " _
95     + vbCrLf + "下記「URL」について複数のコンテンツフィルタに問い合わせました。"
       + vbCrLf + "フィルタに問題があると考えられる場合は「不満を上申」してく
       ださい。" _
96     + vbCrLf + vbCrLf + "【フィルター一覧】" + vbCrLf + "*_Microsoft_ファミリー
       セーフティ" + vbCrLf + "*_Microsoft_ファミリーセーフティ" + vbCrLf +
       "*_Microsoft_ファミリーセーフティ" + vbCrLf + vbCrLf + "
       【URL】" + vbCrLf + TextBox1.Text + vbCrLf + vbCrLf + "【結果】" + vbCrLf
       + "F1:_" + F1HumanReadable + vbCrLf + "F2:_" + F2HumanReadable +
       vbCrLf + "F3:_" + F3HumanReadable + vbCrLf + vbCrLf + "【現在時刻】
       " + vbCrLf + dtNow.ToString() + vbCrLf + "
       =====
       "
97     End Sub
98
99 End Class
```

A.2 ウェブフィルタリングソフトウェア評価ツール

A.2.1 YamakawaNihonshiB-UrlCollector.rb

次のプログラムの作成には、Ruby-1.8.7、「近代日本と世界大戦」『日本史 B 用語集』[23, pp.289-297] 第二次世界大戦節の全ての単語 168 単語^{*1*2}を使用した。

ソースコード A.2 YamakawaNihonshiB-UrlCollector.rb

```
1 # -*- coding: utf-8 -*-
2 # 攻撃コードを含むウェブサイトアクセスする可能性があります。
3
4 require 'rubygems'
5 require 'mechanize'
6
7 agent = Mechanize.new
8 agent.user_agent_alias = 'Windows IE 7'
9
10 # 検索する用語を配列 words として準備
11 words = ["北部仏印進駐",
12 "援蒋ルート",
13 "日独伊三国同盟",
14 "ABCD ライン",
15 "日米交渉",
16 "野村吉三郎",
17 "来栖三郎",
18 "ハル",
19 "ハル = ノート",
20 "グルー",
21 "豊田貞次郎",
22 "日ソ中立条約",
23 "独ソ戦争",
24 "イギリス・ソ連相互援助協定",
25 "大西洋憲章",
26 "関東軍特種演習",
27 "第三次近衛内閣",
28 "南進政策",
29 "南部仏印進駐",
30 "帝国国策遂行要領",
31 "御前会議",
32 "東条英機",
33 "東条内閣",
34 "太平洋戦争",
35 "真珠湾攻撃",
36 "マレー半島上陸",
37 "シンガポール華僑虐殺事件",
38 "翼賛選挙",
39 "翼賛政治会",
40 "大日本政治会",
```

*1 重複した“御前会議〔開戦〕”と“御前会議〔終戦〕”は、“御前会議”に統合した。

*2 “近衛内閣（第三次）”を“第三次近衛内閣”とした。

41 "食糧管理法",
42 "軍需省",
43 "軍政",
44 "軍票",
45 "大東亜共栄圏",
46 "大東亜会議",
47 "自由インド仮政府",
48 "パターン死の行進",
49 "泰緬鉄道",
50 "731部隊",
51 "毒ガス",
52 "満州帝国協和会",
53 "皇民化政策",
54 "南次郎",
55 "皇国臣民の誓詞",
56 "創氏改名",
57 "朝鮮人の強制連行",
58 "従軍慰安婦問題",
59 "中国人の強制連行",
60 "三光作戦",
61 "抗日運動",
62 "反戦同盟",
63 "国民勤労働員令",
64 "国民勤労報国協力令",
65 "勤労働員",
66 "学徒勤労働員",
67 "女子挺身隊",
68 "学徒出陣",
69 "国民義勇隊",
70 "陸軍志願兵制度",
71 "皇民奉仕会",
72 "高砂族",
73 "ミッドウェー海戦",
74 "アッツ島戦",
75 "ガダルカナル島戦",
76 "インパール戦",
77 "マリアナ沖海戦",
78 "サイパン諸島",
79 "レイテ島海戦",
80 "特別攻撃隊",
81 "硫黄島の戦い",
82 "沖縄戦",
83 "鉄血勤皇隊",
84 "ひめゆり隊",
85 "太田実",
86 "牛島満",
87 "対馬丸",
88 "戦艦大和",
89 "本土空襲",
90 "東京大空襲",
91 "学童疎開",
92 "本土決戦",
93 "小磯國昭",
94 "小磯内閣",

95 "最高戦争指導会議",
96 "近衛上奏文",
97 "終戦工作",
98 "カイロ宣言",
99 "ローズヴェルト",
100 "チャーチル",
101 "テヘラン会議",
102 "イタリアの降伏",
103 "ヤルタ協定",
104 "スターリン",
105 "鈴木貫太郎",
106 "鈴木貫太郎内閣",
107 "ドイツ降伏",
108 "ポツダム宣言",
109 "原子爆弾",
110 "ソ連の対日参戦",
111 "シベリア抑留",
112 "無条件降伏",
113 "終戦の詔勅",
114 "降伏文書調印",
115 "重光葵",
116 "日本分割占領計画",
117 "大東亜戦争全史",
118 "戸坂潤",
119 "傾向映画",
120 "五所平之助",
121 "清沢洌",
122 "正木ひろし",
123 "桐生悠々",
124 "吉田富三",
125 "桜田一郎",
126 "釈迢空",
127 "金子光晴",
128 "宮柊二",
129 "亀井勝一郎",
130 "保田与重郎",
131 "皇国史観",
132 "日本文学報国会",
133 "大日本言論報国会",
134 "高見順",
135 "阿部知二",
136 "武田麟太郎",
137 "小林秀雄",
138 "伊藤整",
139 "丹羽文雄",
140 "堀辰雄",
141 "火野葦平",
142 "石川達三",
143 "軽演劇",
144 "エロ・グロ・ナンセンス",
145 "榎本健一",
146 "カジノ＝フォーリー",
147 "古川緑波",
148 "プロレタリア演劇",

```
149 "前進座",
150 "島田正吾",
151 "辰巳柳太郎",
152 "宇野重吉",
153 "杉村春子",
154 "山本安英",
155 "西条八十",
156 "古賀政男",
157 "軍歌",
158 "日本美術報国会",
159 "霞光",
160 "松本竣介",
161 "新人画会",
162 "大東亜戦争美術展",
163 "宗教団体法",
164 "灯台社",
165 "大本教",
166 "ひとのみち教団",
167 "創価教育学会",
168 "横浜事件",
169 "人口政策要項",
170 "金属類回収令",
171 "千人針",
172 "国民服",
173 "木炭自動車",
174 "ルース・ベネディクト",
175 "強制収容所",
176 "満鉄特急あじあ号",
177 "水豊発電所",
178 "銃後"]
179
180 # 配列 words の最後まで一行ずつ取り出す繰り返し処理
181 words.each{|k|
182
183   # Google へ移動
184   agent.get('http://www.google.co.jp/')
185
186   # Google で単語を検索
187   agent.page.form_with(:name => 'f'){|form|
188     form.field_with(:name => 'q').value = k
189     form.click_button
190   }
191
192   # 検索結果 1 ページ分の URL をまとめて取得
193   url = agent.page.root.search('a.l').map{|j|
194     j['href']
195   }
196
197   # すべての URL にアクセスする。
198   # アクセスできない場合は、例外としてエラーメッセージを表示する。
199   url.each{|i|
200     begin
201       agent.get(url)
202     rescue TimeoutError
```

```
203     puts "__Timeout_Error__"
204     rescue Mechanize::ResponseCodeError => exception
205         case exception.response_code
206             when '403' then
207                 puts "__403_Forbidden__"
208             when '404' then
209                 puts "__404_Not_Found__"
210             when '503' then
211                 puts "__503_Service_Unavailable__"
212             else
213                 puts "__Unknown_Exception__"
214             end
215         end
216
217     # アクセス先の URL を表示
218     puts agent.page.uri
219
220     # 10 秒のウェイト
221     sleep 10
222 }
223 }
```

A.2.2 ieLauncher.rb

次のプログラムの作成には、Ruby-1.9.2 (mswin32)、Microsoft Internet Explorer 8.0 を使用した。

ソースコード A.3 ieLauncher.rb

```
1  # -*- coding: utf-8 -*-
2  # 攻撃コードを含むウェブサイトにアクセスする可能性があります。
3
4  # 動作確認用の変数の初期化
5  i = 0
6
7  # 接続する URL を事前に用意する。
8  url = ['http://www.example.com/',
9         'http://www.example.net/',
10        'http://www.example.org/']
11
12 # URL 分繰り返し処理する。
13 url.each{|row|
14     # 動作確認用の番号表示
15     i = i + 1
16     puts i
17
18     # each 文のカウンタ変数を一時変数にコピーする。
19     row_temp = row
20
21     # ブラウザに URL を与えて起動する。
22     pid_ie = Process.spawn('"'C:\Program Files\Internet Explorer\iexplore.exe"' + ' "' +
23                             row_temp + '"')
23 }
```

```

24  # ブラウザの処理の完了を待つ。
25  sleep 10
26
27  # 起動中のブラウザの URL を記録するプログラムを起動する。
28  pid_url = Process.spawn('LastUrlRecorder.exe')
29
30  # 起動中のブラウザの title 要素内の文字列を取得するプログラムを起動する。
31  # pid_title = Process.spawn('LastTitleRecorder.exe')
32
33  # URL と title の記録の完了を待つ。
34  sleep 10
35
36  # ブラウザを終了する。
37  pid_taskkill = Process.spawn('taskkill /IM iexplore.exe /T')
38
39  # ブラウザの終了を待つ。
40  sleep 5
41
42  # プロセスを回収してリソースを解放する。
43  Process.wait(pid_ie)
44  Process.wait(pid_url)
45  # Process.wait(pid_title)
46  Process.wait(pid_taskkill)
47
48  # 動作確認用のステータス表示
49  puts "done"
50 }

```

A.2.3 LastUrlRecorder

次のプログラムの作成には、Microsoft Visual Basic 2010 を使用した。Reference には Microsoft Internet Controls と、Microsoft HTML Object Library を追加した。

ソースコード A.4 LastUrlRecorder (Module1.vb)

```

1 Imports System.IO
2
3 Module Module1
4
5     Sub Main()
6
7         Dim urlList As New ArrayList()
8         Dim shw = New SHDocVw.ShellWindows
9         Dim ie As SHDocVw.ShellBrowserWindow = Nothing
10
11         ' 起動中の Microsoft Internet Explorer から URL を取得する。
12         For Each ie In shw
13             If TypeName(ie.Document) = "HTMLDocumentClass" Then
14                 urlList.Add(ie.LocationURL)
15             End If
16         Next ie
17
18         ' 取得した URL をファイルに書き込む。

```



```
19      System.IO.File.AppendAllText("lasturl.txt", urlList(0).ToString.TrimEnd(" ", "
      *", "#") + vbCrLf)
20
21  End Sub
22
23 End Module
```

A.2.4 LastTitleRecorder

次のプログラムの作成には、Microsoft Visual Basic 2010 を使用した。Reference には Microsoft Internet Controls と、Microsoft HTML Object Library を追加した。

ソースコード A.5 LastTitleRecorder (Module1.vb)

```
1 Imports System.IO
2
3 Module Module1
4
5     Sub Main()
6
7         Dim urlTitle As New ArrayList()
8         Dim shw = New SHDocVw.ShellWindows
9         Dim ie As SHDocVw.ShellBrowserWindow = Nothing
10
11         ' 起動中の Microsoft Internet Explorer のタイトルバーの文字列を取得する。
12         For Each ie In shw
13             If TypeName(ie.Document) = "HTMLDocumentClass" Then
14                 urlTitle.Add(ie.LocationName)
15             End If
16         Next ie
17
18         ' 取得した文字列をファイルに書き込む。
19         System.IO.File.AppendAllText("lasttitle.txt", urlTitle(0) + vbCrLf)
20
21     End Sub
22
23 End Module
```

索引

インターネット, 2, 4, 7, 10, 19
ウェブフィルタリングソフトウェア, 1, 2, 4-10, 14,
15, 21, 24
教科書センター, 12, 14, 15
検定, 2, 5, 12, 15
国民運動, 2, 19
遮断, 1, 2, 4, 6, 7, 9, 10, 14
第二次世界大戦, 7, 24
調査, 4, 7, 20
比較, 1, 5, 6, 10, 14, 15, 21
フィルタリング, 3, 4
有害情報, 1, 2, 4, 19
歴史教科書, 1, 2, 4-6, 14, 15, 18
歴史教科書問題, 1, 2, 15