

ウェブブラウザの種類とバージョンに着目した HTML-lint の提案と試作

大萬 将陽[†]
明治大学大学院理工学研究科[†]

水村 総之介[‡]
明治大学理工学部[‡]

早川 智一[§]
明治大学理工学部[§]

1. はじめに

近年、HTML (HyperText Markup Language) の仕様からバージョンの概念がなくなった。これは、HTML の標準仕様が HTML Living Standard (以下、HLS) [1] になり、仕様が逐次変更されるようになったからである。

これにより、HTML の要素・属性とウェブブラウザ (以下、ブラウザ) の種類・バージョンとの対応状況 (以下、HLS への対応状況) の把握が困難になった。なぜならば、HLS の逐次的な変更に伴い、HLS への対応状況も不定期に変更されるからである。これにより、開発者がブラウザの対応状況を把握するために、HLS への対応状況を逐一確認する必要がある。

この困難さに対し既存システムを用いることで緩和することはできるが、充分ではない。なぜならば、既存システムは HTML 文書全体の判別が容易ではないからである。既存システムは 1 つ 1 つの要素・属性を確認する必要があるため、必要な工数が増大してしまう。

本論文では前述の困難を解決するために、HTML 文書全体の HLS への対応状況を一括して判別する HTML-lint を提案する。これにより提案 lint の利用者は、少ない工数で HLS への対応状況を確認できる。

本論文の構成は次のとおりである。2 章では関連技術を紹介する。3 章では提案手法を概説する。4 章・5 章では設計・実装を説明する。6 章では評価結果を報告する。7 章では今後の展望を述べる。

2. 関連技術

Can I use [2] は、HTML の要素や属性を 1 つ入力することで、HLS への対応状況を視覚化するウェブサイトである。例えば a 要素の対応状況を確認する場合、入力欄に「a」と入力すると、各ブラウザのバージョン毎の HLS への対応状況が

画面に表示される。同サイトは HLS への対応状況を判別できる点で提案 lint と類似するが、個々の要素・属性しか判別できない点で異なる。

3. 提案手法

3.1 提案手法の概要

図 1 に提案 lint の概要を示す。提案 lint は CLI (Command Line Interface) で動作させることを想定する。これは、CI (Continuous Integration) で動作可能にするためである。提案 lint の利用の流れは以下のとおりである：(1) 利用者はブラウザの種類とバージョンを 1 つずつ指定し、確認対象の HTML 文書を提案 lint に入力する；(2) 提案 lint は入力された HTML 文書中の要素・属性が指定したブラウザの種類・バージョンの HLS への対応状況に反していれば指摘する；(3) 指摘内容を入力する。

提案 lint の特徴は、HLS への対応状況を独立のファイルとして保持・更新できる点である。これにより、HLS の更新の追従が用意になる。具体的には、提案 lint はソースコードを変更せず、情報ファイルを更新するだけで追従ができる。

3.2 提案 lint の指摘事項

提案 lint は以下の 3 つの観点で指摘を行う：(1) 用いることが非推奨である要素・属性；(2) 指定されたブラウザとバージョンに未対応な要素・属性；(3) 指定されたブラウザとバージョンに対応しているかどうか不明な要素・属性。これにより開発者は、作成した HTML 文書のどの要素・属性が HLS への対応状況に反しているか確認できる。

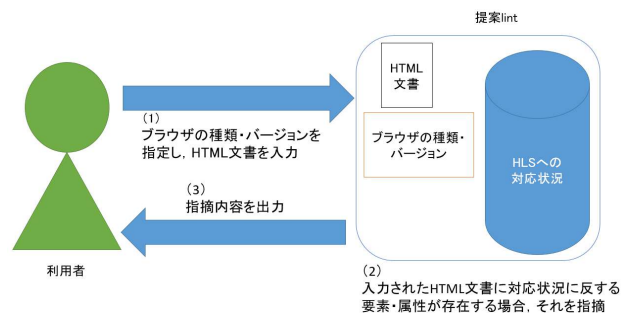


図 1 提案 HTML-lint の概要

A proposal of HTML-lint based on web browser information

[†]Masataka Daiman, Meiji University Graduate School

[‡]Sonosuke Mizumura, Meiji University

[§]Tomokazu Hayakawa, Meiji University

```
$ NODE_PATH=src/ node src/load-html.js \
> -f index.html -b chrome -v 87.0.4280.66
Browser name: chrome
Browser version: 87.0.4280.66
Analysing now ...
a.shape: deprecated attribute.
a.shape: unsupported attribute.
```

図2 提案 lint の実行例 (スクリーンショット)

3.3 提案 lint の利用例

図2は提案 lint の実行例である。具体的には、提案 lint に Google Chrome (以下, Chrome) の最新バージョンを指定し, 自作の HTML 文書を入力した結果である。入力した HTML 文書は Chrome の最新バージョンに対応していない属性—a 要素の shape 属性—を含む。図2より, 提案 lint が前述の属性に対し, (1) 非推奨であること, (2) 指定したブラウザのバージョンでは未対応であること—を指摘していることが分かる。

4. 設計

図3に提案 lint の概要ユースケース図を示す。提案 lint 内部の処理の流れは次の通りである：(1) 入力された HTML 文書を DOM (Document Object Model) 木に変換し, 木構造を深さ優先で探索しながら, 以降の処理を行う；(2) DOM 木内の要素・属性が, 指定されたブラウザの種類・バージョンの HLS への対応状況に反しているかを判別する；(3) 発見した内容に対応した指摘内容を入力する。

5. 実装

提案 lint を実装するために, 表1のソフトウェアを用いた。JavaScript 実行環境に Node.js [3] を用いた。同実行環境内での DOM 操作の実現のために jsdom [4] を用いた。HLS への対応状況の判別に browser-compat-data [5] を用いた。

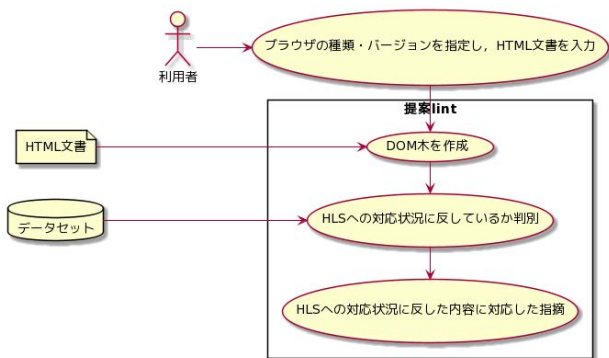


図3 提案 lint の概要ユースケース図

表1 提案 lint の実装に使用したソフトウェア

ソフトウェア	バージョン	備考
Node.js	14.15.3	JavaScript 実行環境
jsdom	16.4.0	DOM 操作ライブラリ
browser-compat-data	3.0.1	HLS への対応情報

表2 HLS への対応状況の調査に要した時間

ウェブサイト\システム	Can I use	提案 lint
Google	16分 24秒	16秒
YouTube	8分 11秒	24秒
Tmall	20分 04秒	13秒

6. 評価

提案 lint の評価として, HLS への対応状況の確認が既存システムと比較して容易になっているかを確認した。具体的には, Alexa Top Sites [6] の上位3位 (Google・YouTube・Tmall) の HTML 文書を用いて, Chrome の最新バージョンに対応していない要素・属性を Can I use と提案 lint とで調査し, 調査にかかった時間を比較した。なお, 提案 lint は Can I use と同じデータセットを用いているため, 両者の正確性については同等である。

計測結果を表2に示す。表2より, 提案 lint が Can I use よりも調査にかかった平均時間が約97.5%減少したことが分かる。Can I use の調査時間が長いのは, 検査する HTML 文書のどの要素・属性が HLS への対応状況に反するか不明なため, 文書を先頭から確認する必要があるからである。一方, 提案 lint は一括して判別できるため, 調査時間が削減される。よって, HLS への対応状況の確認が既存システムよりも容易であることを確認した。

7. おわりに

本論文では HLS への対応状況に着目した HTML-lint を提案した。提案 lint を試作した結果, HTML 文書の対応状況の判別にかかる工数が既存システムと比較して減少することを確認した。

今後の展望として, 提案 lint の判別能力の向上を行うとともに, 詳細な評価を行う予定である。

参考文献

[1] WHATWG: HTML Living Standard, <https://html.spec.whatwg.org/multipage/>.
 [2] Deveria, A.: Can I use, <https://caniuse.com/>.
 [3] OpenJS Foundation: Node.js, <https://nodejs.org/>.
 [4] jsdom: jsdom, <https://www.npmjs.com/package/jsdom/>.
 [5] MDN: browser-compat-data, <https://github.com/mdn/browser-compat-data/>.
 [6] Alexa: The top 500 sites on the web, <https://www.alexa.com/topsites/>.