

Exploring the Use of XPath Queries for Automated Assessment of Student Web Development Projects

Russell Thackston
Department of Information Technology
Georgia Southern University
Statesboro, Georgia USA
rthackston@georgiasouthern.edu

ABSTRACT

Web development courses are core to Information Technology (IT) and Computer Science (CS) curriculums. As a result, hundreds of students each semester enroll in these courses to learn HTML, CSS, and JavaScript. This burdens faculty with the need to grade tens or hundreds of web assignments each semester. Few tools exist to automate the grading of such assignments. This research explores the use of XPath queries, which may be used to evaluate HTML documents due to their loose relationship to XML. A web application was constructed to graphically define XPath queries, then provided to web development faculty to regrade past assignments. Based on faculty feedback, it was determined that XPath queries are capable of limited grading of HTML document against past assignments. However, by designing new assignments tailored for automated grading and by enhancing the web application with additional features, it is clear a significant portion, if not all, of the HTML documents for a web development assignment may be automatically graded.

CCS CONCEPTS

- Information systems~World Wide Web
- Applied computing~Education
- Social and professional topics~Professional topics~Computing and business~Automation

KEYWORDS

HTML; XPath; Automation; Education; Web Development

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

SIGITE '20, October 7–9, 2020, Virtual Event, USA
© 2020 Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-7045-5/20/10...\$15.00
<https://doi.org/10.1145/3368308.3415389>

ACM Reference Format

Russell Thackston. 2020. Exploring the Use of XPath Queries for Automated Assessment of Student Web Development Projects. In Proceedings of SIGITE '20 conference, Virtual Event, USA, October 2020, 5 pages. DOI: 10.1145/3368308.3415389.

1 INTRODUCTION

The Association for Computing Machinery (ACM) 2017 curriculum guidelines list Web and Mobile Systems as essential domain knowledge for any Information Technology (IT) program. [1] Additionally, the ACM guidelines recommend Platform Based Development/Web Platforms as an elective in Computer Science (CS) programs. [2] And, while the ACM guidelines for Information Systems and Software Engineering do not specifically recommend web development courses, web technologies are repeatedly referenced.

Unsurprisingly, many IT and CS programs offer introductory courses in web technologies. A survey of the last six schools to host SIGITE revealed each of them offered an introductory web page development course. Three of the institutions provided enrollment statistics on-line, showing multiple course sections being offered in support of 25-40 students per section. Enrollment for the remaining three institutions was not available on-line. Georgia Southern University regularly offers multiple sections of their web development courses for both majors and non-majors.

School	Course	Sample Enrollment
Univ. of Nebraska	CIST 1300	123 (Fall 2019)
Univ. of Washington	TINFO 230	-
Broward College	ISM 3054C	-
Rochester Inst. of Tech.	ISTE 140	93 (Spring 2020)
Univ. of Massachusetts	IT 240	75 (Fall 2019)
DePaul Univ.	IT 231	-
Georgia Southern Univ.	IT 1430/1230	154 (Fall 2019)

2 RATIONALE

Extrapolating from the sample enrollment data, it's clear that hundreds – perhaps thousands – of students enroll in a web development course each semester in the U.S. university system. These courses teach core web technologies, such as Hypertext Markup Language (HTML), Cascading Style Sheets (CSS), and JavaScript, using labs, projects, homework, and exams. As with most programming assignments, a high degree of hand grading of

the students' source code is required to properly assess each assignment.

This is primarily because HTML offers multiple different ways to accomplish a task. For example, an instructor may ask a student to bold a sentence using a `` tag, but the student could also accomplish the task using `` tag or with ``. All three will render the same on the web page. Furthermore, small, syntactical errors on a student's part can result in significant visual errors on the rendered page. For example, an image tag may be included in an HTML document with every attribute correct except the source (i.e. "src=" vs "source="). Without looking at the source code, an instructor cannot ascertain what was wrong or if partial credit is due.

A survey of the rubrics used in Georgia Southern University's web courses demonstrates the level of detail required to evaluate an HTML assignment. The rubrics describe specific tags that should be used, valid nesting of tags, encoding of HTML entities (e.g. `©` for the copyright symbol ©), use of semantic tags, linking to external files, etc. Each rubric surveyed listed twenty or more specific criteria to check. Since these courses usually include five lab and/or homework assignments, each semester the instructor is hand grading 15,000 specific elements (20 criteria per assignment, 5 assignments per student, and 150 students per semester).

Recent events have amplified the challenges in grading web assignments. The Covid-19 pandemic has resulted in significant budget reductions to Georgia universities, has directly impacted the web instructors, who have lost their teaching assistants for the upcoming fall 2020 semester. [3] It appears likely that similar impacts will be felt at schools across Georgia and the United States. Given the sheer volume of work and lack of resources, the need for automated grading is clear.

3 LITERATURE REVIEW

While many approaches exist for the automated grading of programming assignments – Xunit approaches, acceptance testing frameworks, and web testing frameworks – few tools are specifically tailored to the nuances of web development assignments. [4] WebWolf presented a promising solution, as a "simple framework for [the] automated assessment of webpage assignments in an introductory web programming class." However, WebWolf tests are created using XUnit-style code, can only be run locally on the instructor's computer, and the software does not appear to be publicly available for download or use.

Commercial and open source tools, such as JMeter and Postman, can be used to test a website's functionality. [5,6] However, such tools are focused on the rendered website and its behavior, not the source code. For example, JMeter can, through custom user scripting, determine if a specific element exists in the source code of a web page. However, JMeter tests must be run locally, on demand, by the instructor and requires each student's code to be

hosted on a web server. Postman, through the use of cheerio, supports XUnit-style tests of web pages, but also can only be run by the instructor and requires hosting of student assignments on a web server. [7]

4 HTML, XML, XHTML, AND XPATH

HTML is a markup language "primarily designed as a language for semantically describing scientific documents, although its general design and adaptations over the years have enabled it to be used to describe a number of other types of documents." [8] XML is a subset of standard generalized markup language (SGML) that bears many similarities to HTML. [9] In 2000, XHTML was proposed as an SGML to blend HTML semantics with a more rigid XML structure. [10] However, the latest HTML5 specification does not require the rigid XML structure for HTML documents. Development of a separate XHTML specification has been abandoned with the HTML5 specification simply referencing XML-formatted HTML documents as an option.

Despite the fact that HTML5 and XML are separate languages, certain XML-based tools may be used in conjunction with HTML5. For example, the XML Path (XPath) language can be used to query the contents of an HTML Document Object Model (DOM). [11] The XPath query `"/html/head/title[text()]"` will return the title of a web page. The XPath query `"//div"` will return a list/array of all `<div>` elements within a web page, regardless of their nesting within other elements.

5 RESEARCH QUESTION AND METHOD

5.1 Web Application Design

Given that web development assignments are graded based on their content and HTML documents may be searched using XPath, this research seeks to explore the feasibility of pairing the two in student web development projects. A series of twenty assignments over a period of four semesters were reviewed to determine the common types of criteria used within the rubrics to assess student performance (e.g. a specific tag being correctly used in the document, a specific tag contains certain text, etc.).

Based on these generalized criteria, a prototype web application was developed to allow an instructor to define specific criteria rules to evaluate student assignments. The web application first asks the instructor to upload a complete and accurate assignment (i.e. HTML file). The document is parsed into a DOM and rendered on the screen using a tree view. See Figure 1.

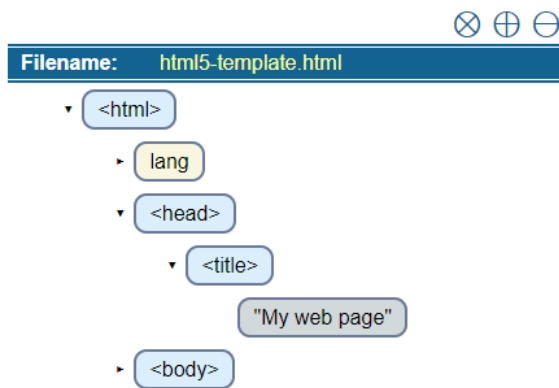


Figure 1: HTML DOM rendered using a tree view.

Using the tree view, instructors would select node(s) to include in the grading process. For example, if the instructor required the web page to be titled “My web page,” they would select the “My Web Page” text node under the <title> node and a new criterion would be added to the rubric with an associated XPath query. By implementing a graphical user interface to define XPath queries, syntax errors could be avoided, and rubrics could be quickly built.

To evaluate a student assignment, the instructor would upload each student’s HTML document and the web application would run the XPath queries against the document. One or more matching nodes returned by the XPath query would indicate the student completed the requirement. Conversely, failure to find any matching nodes would indicate the requirement was not fulfilled. In this manner, an instructor could check for the existence of HTML tags, attributes, attribute values, and text elements.

5.2 Validation

Using the web application, student submissions for the four-semester period would be regraded. The grades assigned by the web application would be compared to the grades assigned by the instructors/teaching assistants. In addition, qualitative data would be collected via interviews with the instructors regarding the benefits and drawbacks of grading via the web application.

6 RESULTS

It became immediately apparent that the web application, in its current form, could not fully replace hand grading. The reasons cited by the instructors included: alignment issues with rubrics; lack of flexibility; and incomplete coverage.

However, the instructors noted the following positive aspects: speed of defining rubrics.

6.1 Alignment Issues

The original rubrics developed for the web development courses assumed the instructor would be hand grading the assignment. Therefore, no effort was made to the design an “automation-friendly” rubric. This resulted in an inability to codify many of the rubric criteria into XPath queries.

For example, all the assignments required the student to include their name in the title of the web document, followed by a hyphen and the assignment name. This requirement was cited as being highly important to the instructor during grading, so they could identify which student’s work was being viewed during grading. The web application had no way of knowing the student’s name and, therefore, could not fully evaluate that criteria. The best the instructors could do was search for the assignment name as a substring of the title.

The instructors indicated that most of these issues could be eliminated by creating new XPath-friendly requirements for each assignment. However, this would require a significant upfront investment and would likely need to be further refined over time.

6.2 Lack of Flexibility

As designed, the web application performed a simple true/false evaluation based on a non-zero number of nodes being returned from the XPath query. However, several requirements in the assignment contained language such as “using 2 or more paragraphs.” The instructors requested they be able to create a criterion which specified a minimum or specific number of nodes to be returned by the XPath query.

Another area cited by the instructors as lacking flexibility related to the automated generation of the XPath. By design, the web application allowed an instructor to select a node to be graded (child) and zero to many parent nodes to also check for. See Figure 2. For example, the instructor could select the “My paragraph” text node as the child and the <html> tag as the parent, resulting in the following an XPath query.

```
//html//body//div[text() [contains(., "My paragraph")]]
```

This XPath query would check for the text inside a <div> inside the <body> and so on. However, it did not allow the instructor to specify attributes of the parent nodes, something common in rubrics. For example, an assignment might require that the “My paragraph” text appear inside a <div> with a specific id attribute (i.e. <div id='leader'>).

This led to the instructors requesting the ability to directly edit the generated XPaths, after graphically defining them.

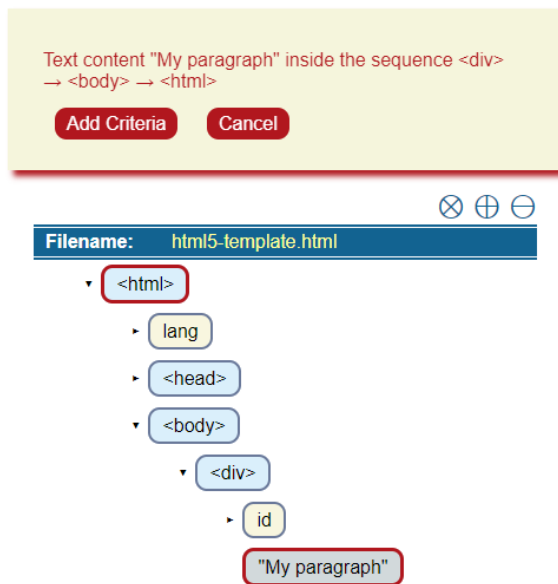


Figure 2: Specifying parent and child nodes.

6.3 Incomplete coverage

Although HTML and XML share many characteristics, there are many elements of web documents that cannot be checked with an XPath. These elements occasionally show up as requirements in the student assignments, resulting in an incomplete coverage of criteria by the web application.

For example, every HTML document turned in by students must contain an HTML comment at the top with the student's name, date, assignment name, etc. Since HTML comments are not valid XML elements, the DOM parser will ignore them, and they cannot be checked with XPath queries.

One instructor suggested this could be remedied by switching from HTML comments to HTML meta tags, such as author, description, and keywords. However, this would require additional upfront investment of time and effort.

Another observation was that web assignments are composed of more than just HTML documents. Most web development assignments quickly grow to include CSS and JavaScript elements, which the web application does not support.

6.4 Partial Credit

When defining the rubric criteria in the web application, it became apparent there was not a one-to-one mapping of requirements to XPath queries without sacrificing the ability to give partial credit.

For example, if an assignment required the student to include an `` tag with an `alt` attribute, then multiple XPath queries would need to be created: one to check for the `` tag, another to check for the `src` attribute, and another to check for the `alt`

attribute. Each of these could layer together to offer partial credit to students who completed some of the requirements, but not all.

However, instructors noted a flaw with the preceding example. If a student did everything right but misnamed the `` tag as `<image>` they would receive no partial credit.

6.5 Rubric Development Speed

Overall, the instructors were satisfied with the speed with which criteria could be defined using the graphical interface. Although all the instructors involved in the research had prepared solutions on-hand to upload to the web application, it was noted that this might not be the case for all web development instructors. Instructors who previously created only a requirements document would now need to also complete the assignment itself before grading criteria could be defined in the web application.

6.6 Assignment Submission

Instructor feedback regarding the grading process was also positive, as the web application clearly listed the successful and missed criteria. However, instructors were quick to request that student be given the ability to submit their own assignments.

6.7 Accuracy

It's clear that the web application is unlikely to make a grading error, provided the defined XPath queries are accurate, whereas a human grader may make the occasional mistake. For example, the researchers discovered a few examples of student work which had been given full credit on a requirement but contained one or more small, difficult to notice errors.

6.8 Scalability

Given a single instructor, hand grading of assignments scales linearly with the number of student submissions. Automated grading also scales linearly with the number of submissions, however the time spent on individual submissions may be reduced.

True scalability may be achieved if students can directly submit their assignments via a web portal, a feature requested by all instructors involved. However, such an enhancement would require additional effort on the part of the instructor to define clear feedback text for each criterion and the submission/resubmission process.

7 CONCLUSIONS

This research demonstrates that XPath queries have the potential to assist in the grading of HTML student projects. However, this potential is hampered by a lack of flexibility, incomplete coverage, and difficulty in offering partial credit. In addition, using XPath queries may require a complete redesign of existing assignments, as well as a significant upfront investment of time and effort to prepare define the queries. This investment, however, could result

in significant improvements to grading in terms of accuracy and scalability.

8 FUTURE WORK

The web application, as designed, has clear potential as well as room for much improvement. Some of the following enhancements were suggested by the instructors and some by the researchers.

New features to implement include self-submission of assignments by students; node counting versus node existence, to support requirements such as “three `div` tags”; wildcards and variables in XPath queries, such as student name, today’s date, numeric value, etc.; plagiarism checking across the class; a

gradebook for instructors; rubric and criteria management; W3C validation of HTML documents.

The web application may also be enhanced to determine if CSS and JavaScript document may be graded against a rubric.

ACKNOWLEDGMENTS

The authors would like to thank the web development faculty of Georgia Southern University for sharing their rubrics, assignments, input, and time.

REFERENCES

[1] ACM. Curriculum Guidelines for Baccalaureate Degree Programs in Information Technology: Retrieved from <https://www.acm.org/binaries/content/assets/education/curricula-recommendations/it2017.pdf>.

[2] ACM. Curriculum Guidelines for Undergraduate Programs in Computer Science: 2013. Retrieved from https://www.acm.org/binaries/content/assets/education/cs2013_web_final.pdf.

[3] University System of Georgia. Board of Regents Provides Authority for Potential Reductions Due to COVID-19. Retrieved from https://www.usg.edu/news/release/board_of_regents_provides_authority_for_potential_reductions_due_to_covid_1.

[4] Ihantola, P. et al. 2010. Review of recent systems for automatic assessment of programming assignments. Proceedings of the 10th Koli Calling International Conference on Computing Education Research (New York, NY, USA, Oct. 2010), 86–93.

[5] Apache Foundation. Apache JMeter. Retrieved from <https://jmeter.apache.org/>.

[6] Postman. The Collaboration Platform for API Development. Retrieved from <https://www.postman.com/>.

[7] Postman. jQuery replaced by CheerioJS in Postman Sandbox. Retrieved from <https://blog.postman.com/jquery-replaced-by-cheeriojs-in-postman-sandbox/>.

[8] W3C. Introduction – HTML5. Retrieved from <https://dev.w3.org/html5/spec-LC/introduction.html>.

[9] W3C. Extensible Markup Language (XML) 1.0 (Fifth Edition). Retrieved from <https://www.w3.org/TR/2008/REC-xml-20081126/>.

[10] W3C. XHTML 1.0: The Extensible HyperText Markup Language (Second Edition). Retrieved from <https://www.w3.org/TR/xhtml1/>.

[11] W3C. XML Path Language (XPath) 3.1. Retrieved from <https://www.w3.org/TR/xpath-31/>.