

Detecting Violations of CSS Code Conventions

Boryana Goncharenko

boryana.goncharenko@gmail.com

July 15, 2015, 11 pages

Supervisor: Vadim Zaytsev
Host organisation: University of Amsterdam



UNIVERSITEIT VAN AMSTERDAM
FACULTEIT DER NATUURWETENSCHAPPEN, WISKUNDE EN INFORMATICA
MASTER SOFTWARE ENGINEERING
<http://www.software-engineering-amsterdam.nl>

Contents

Abstract	2
1 Introduction	3
2 Background	4
2.1 Code conventions and style guides	4
2.2 Ontological analysis	4
3 Evaluating the Need for CSS Code Conventions	5
3.1 Research Method	5
3.2 Results	5
3.3 Analysis	5
4 Discovering Existing CSS Code Conventions	6
4.1 Research Method	6
4.2 Results	6
4.3 Analysis	7
5 Expressing CSS Code Conventions	8
5.1 Analysis of conventions corpus	8
5.2 Abstract syntax	9
5.3 Concrete syntax	9
5.4 Mapping concrete and abstract syntax	9
5.5 Validation	9
5.5.1 Ontology design	9
5.5.2 Ontological analysis	9
6 Conclusion	10
Bibliography	11

Abstract

This section summarises the content of the thesis for potential readers who do not have time to read it whole, or for those undecided whether to read it at all. Sum up the following aspects:

- relevance and motivation for the research
- research question(s) and a brief description of the research method
- results, contributions and conclusions

Chapter 1

Introduction

Code conventions put constraints on how code should be written in the context of a project, organization or programming language. Style guides can comprise conventions that refer to whitespacing, indentation, code layout, preference of syntactic structures, code patterns and anti- patterns. They are mainly used to achieve code consistency, which in turn improves the readability, understandability and maintainability of the code [Citations].

Style guides are often designed in an ad hoc manner. Coding conventions typically live in documents that contain a description of each rule in natural language accompanied by code examples. This is the case with the style guidelines of Mozilla [8], Google [6], GitHub [10], WordPress [11] and Drupal [4]. To apply the conventions, developers first need to read, understand and apply them manually. Such an approach introduces a number of issues. Using natural language can make guidelines incorrect, ambiguous, implicit or too general. Another problem is that developers apply conventions manually, which increases the chances of introducing violations involuntarily. There are tools that check for compliance with guidelines, however, they are often hard to customize or limited to one type of violations, e.g. only whitespacing.

The core idea behind the project is to provide a solution that lets developers express an arbitrary set of coding conventions and detect their violations automatically in an IDE. Writing conventions in an executable form could assist authors in detecting incorrect, ambiguous or inconsistent guidelines. Automatic detection of violations could minimize the effort required by developers to write code that complies to the guidelines. To meet the constraints of a Masters project, the implementation is limited to the domain of Cascading Style Sheets (CSS). The project requires determining the need for CSS code conventions in organizations, collecting and analyzing available style guides, and providing a way to express conventions. Specifically, the project attempts to answer the following set of questions:

- **Research Question 1:** Do developers still maintain plain CSS?
- **Research Question 2:** What code conventions for CSS exist?
- **Research Question 3:** How to express existing CSS code conventions?

The thesis is organized as follows. Chapter 2 contains background notions and terms used throughout the thesis. The method and results for RQ1 are presented in Chapter 3. Discovering the existing code conventions is illustrated in Chapter 4. The design and description of a DSL is presented in Chapter 5. Chapter 6 concludes the thesis.

Chapter 2

Background

2.1 Code conventions and style guides

2.2 Ontological analysis

Ontology: “explicit specification of a conceptualization” “[Gruber <http://tomgruber.org/writing/onto-design.pdf>]

Top-level ontology: “the general (domain-independent) core of an information systems ontology” “[Milton <http://ontology.buffalo.edu/medo/Bunge-Chisholm.pdf>]

Domain-specific ontology: “the extension or specification of a top-level ontology with axioms and definitions pertaining to the objects in some given domain. ”

Chapter 3

Evaluating the Need for CSS Code Conventions

I will talk about answering the first research question.

3.1 Research Method

Despite the new features added in the second [1] and third [5] versions of CSS, the language has obvious limitations, e.g. lack of variables. A number of preprocessors have evolved to tackle the downsides of CSS. Solutions such as SASS [3], LESS [9] and Stylus [7] offer enhanced or even different syntax and translate it to CSS. Preprocessors are not only ubiquitously recommended, but also widely adopted in practice. Obviously, using such solutions avoids the need for CSS code conventions because the code is generated and not maintained directly. Thus, CSS code conventions make sense only if developers handcraft the CSS files.

To determine whether CSS is written and maintained as opposed to being generated, all commits to open source repositories hosted on GitHub for 2015 (up to April) are being analyzed. If the commit contains a file with extension .scss, .sass, .less or .styl, it is considered preprocessor maintenance. In case the commit contains files with the .css extension and no preprocessor extensions, it is considered maintenance of CSS. To exclude cases in which developers commit third-party CSS, only commits that modify CSS files are taken into consideration. Commits that add or delete CSS files are ignored.

3.2 Results

Despite of the popular belief that nowadays preprocessors are prevailing than CSS, results illustrate plain CSS is still used. Figure 1 summarizes the findings.

Having the above in mind I assume that CSS is still maintained in practice.

3.3 Analysis

What if the main users of preprocessors are private?

What if the time interval was too short?

What if there are other preprocessor that are not counted?

Chapter 4

Discovering Existing CSS Code Conventions

4.1 Research Method

The CSS community has produced a pool of recommendations, best practices, and style guides, but how to choose the among them? Since the primary organization responsible for the specification of CSS has not recommended code conventions, any selection strategy based on the author of the conventions could be considered cherry picking.

To determine the set of code conventions, two searches with the keywords CSS code conventions have been made using the search engines <http://duckduckgo.com> and <http://google.com>. The first 50 results of each search have been analyzed. From each result only conventions about pure CSS are taken into account and guidelines for CSS preprocessors are ignored. In case the result is an online magazine or a blog post that links to other resources, these references are considered as results and analyzed separately.

Discuss problem with conventions here:

Overgeneralized conventions - The description of the convention is too general to be applied, e.g. dont use CSS hackstry a different approach first.

Incorrect conventions - There is a discrepancy between the description of the rule and the provided example. An instance of such contradiction is when the convention nothing but declarations should be indented is followed by a code snippet illustrating that rules in media queries should also be indented.

Ambiguous conventions - There is more than one interpretation of a convention. For example, rules with more than 4 selectors are not allowed could be seen as forbidding multi- selectors with more than four selectors, or disallowing selectors with more than four simple- selectors.

Implicit conventions - There are rules that are not explicitly stated and could only be inferred by the other rules. For example, the convention you can put values on multiple lines is not preceded by a convention that requires values to appear on one line.

Inconsistencies between conventions in one ruleset - Google say that charsets should not be used in css, but later say that in charsets double quotes should be used. Surely, this opens the question - so when do I need to use charsets?

4.2 Results

Results of the two searches include the CSS coding guidelines of CSS professionals as well as leading companies, e.g. Google, Mozilla, GitHub, Wordpress. The accumulated corpus consists of 165 unique coding conventions. . Here is an example:

A full list is available at [this GitHub page](#).

4.3 Analysis

There are conventions that cannot be detected at all?

What if not everyone have published their style guides? Well, some respectful companies did.

Chapter 5

Expressing CSS Code Conventions

5.1 Analysis of conventions corpus

Code conventions is an umbrella term that comprises a vast set of rules. To check what the CSS conventions are all about, I analyzed the conventions and it turned out there are three broad groups of conventions. In general, CSS conventions refer to whitespacing and indentation, to syntax preference or to programming style.

Further, the analysis consists of analyzing each individual convention.

The approach used to explore the domain consists of analyzing how the current system works. In other words, the analysis aims at revealing how developers detect violations manually, what is knowledge developers need and what are the particular steps they make. The domain analysis phase consists of two steps:

- Determine possible violations. In this step the meaning of the convention is discussed and violations are made explicit.
- List the specific actions that need to be taken to detect violations manually. Developers check their code for compliance manually. To perform such checks, developers need to understand different concepts, e.g. the concept of a rule, html element, ids, etc and perform certain actions, such as find a structure, evaluate a constraint etc.

The following analysis of three conventions illustrates the process described above. Analysis of all conventions in the corpus are available at [GitHub](#).

Convention: Disallow empty rules.

Author: CSS lint

Violations: Presence of rulesets that do not contain declarations. In case at least one declaration is present, the ruleset does not violate the convention. Examples include:

```
1 .myclass { } /* violation */  
2 .myclass { /* Comment */ } /* violation */  
3 .myclass { color: green; } /* not violation */
```

Actions: Recognize rulesets and declarations. Determine whether a ruleset does not contain any declarations.

This convention aims at getting rid of one type of refactoring leftovers - rulesets without declarations [CSS lint]. Removing empty rulesets reduces the total size of CSS that needs to be processed by the browser. One possible approach for discovering violations of the convention at hand is to search the

stylesheet for rulesets and then check whether each ruleset contains a declaration. To perform this search successfully, developers need to understand the concept of a ruleset and a declaration, i.e. they need to be able to recognize these two CSS structures. Further, developers need to determine relations between structures, particularly, whether a ruleset contains a declaration.

In this way, all conventions have been analyzed and as a result, the following set of conclusions have been made:

- 1) Every convention imposes a set of constraints over the program. Violations occur when these constraints are not met.
- 2) Conventions can refer to nodes in abstract syntax tree, concrete syntax tree and parse tree of CSS.
- 3) Conventions can refer to nodes based on their type or function in the program (hex, color)
- 4) Conventions can refer to nodes based on CSS-specific knowledge, e.g. is vendor-specific property
- 4) Conventions can refer to nodes based on their context. Typically, conventions do not target a node in isolation but a number of related nodes that form a pattern. The nodes do not have to be immediately related, they could be scattered across the tree.
- 5) Conventions could be related. For example, it is often the case that two conventions are related within the boundaries of a style guide. Often a given convention serves as an exception for another convention, or relaxes the constraints of another convention.

5.2 Abstract syntax

Here is a big diagram of my abstract syntax

5.3 Concrete syntax

Here is my grammar

5.4 Mapping concrete and abstract syntax

Here is how they map

5.5 Validation

Why ontologies are helpful in DSL development [<http://ceur-ws.org/Vol-395/paper02.pdf>]

5.5.1 Ontology design

5.5.2 Ontological analysis

You can refer to pretty much anything (websites, blog posts, software) through `misc` type of entry [[Par08](#)]:

Chapter 6

Conclusion

This work makes several contributions. 1) 2) It contains a summary of existing CSS coding conventions. 3) It designs a domain specific language that expresses conventions and its interpreter to detect violations automatically.

Bibliography

- [Par08] Terence Parr. ANTLR—ANother Tool for Language Recognition. Toolkit website, 2008.
URL: <http://antlr.org>.