Designing Scalable Syntaxes for Interface-Modifying Digital Self-Control Tool Interventions

R.X. SCHWARTZ, University of Virginia, United States
PANAGIOTIS APOSTOLELLIS, University of Virginia, United States

How can interface-modifying digital self-control tools "scale up" so that they can become widely used, adaptable, and reliable systems in a user's digital lifestyle? This paper is motivated by ad blockers and password managers, two types of popular user interface modification tools (outside of the digital self-control area) which provide reliable user experiences among a large scale of websites. We apply a framework based on the syntaxes of ad blockers and password managers to the digital self-control area, and discuss the current state of the digital wellbeing field in terms of progress towards scalable interface modifications. We also propose benefits and potential limitations of interface-modifying syntaxes for digital self-control tool interventions.

CCS Concepts: • **Software and its engineering** → **Interface definition languages**; • **Human-centered computing** → *Interaction paradigms*.

Additional Key Words and Phrases: syntax, digital wellbeing, digital self-control tool

ACM Reference Format:
R.X. Schwartz and Panagiotis Apostolellis. 2022. Designing Scalable Syntaxes for Interface-Modifying Digital Self-Control Tool Interventions. In *Advanced Visual Interfaces (AVI) 2022, Designing for Meaningful Interactions and Digital Wellbeing (Design4DWB) Workshop*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.18130/bv65-yw11

1 INTRODUCTION

Digital self-control tools, otherwise known as DSCTs, are "self-binding applications that constrain future usage of devices or specific applications" [1]. These tools have been developed both in academia and in industry for a variety of platforms, including iOS, Android, and Google Chrome [2]. A single DSCT can have several interventions, or features that attempt to change user behavior.

A proportion of DSCTs influence user behavior by creating a new, standalone intervention that is paired with the targeted website or application. These interventions are called *external interventions*, or those which do not directly modify the interface of the website or application [3]. Examples include time limits, usage visualizations, and goal-setting [2]. Another proportion of DSCTs directly modify website or application interfaces in attempts to control usage, including by changing the size or appearance of elements [2]. These interventions are called *internal interventions* or *interface-modifying interventions*, and "make the original experience easier to manage" [3] [4]. External and internal DSCTs exist on a spectrum: a DSCT could be considered increasingly internal the more it customizes the interface of a website or application [3]. One drawback of internal (interface-modifying) DSCTs is that the tool must possess specialized knowledge about an interface in order to successfully modify it. In contrast, purely external DSCTs (e.g. app-agnostic timer or blocking DSCTs) may be applied regardless of the interface at hand [3].

Although previous literature has expressed the desirability of modifying interfaces at scale using internal DSCTs [5], internal DSCTs are usually designed on an interface-specific basis (i.e. YouTube or Twitter) and are unable to make all interventions scale up to all interfaces [3] [4] [6]. This interface-specific focus prevents internal DSCTs from easily adapting to new environments, such as previously unseen interfaces or updates to existing interfaces. As a result, internal DSCTs often require increased development time, prototyping, testing, and maintenance.

In this paper, we suggest that lessons can be learned for "scaling up" internal (interface-modifying) DSCTs by turning to other software which modifies interfaces at scale. Notably, password managers and ad blockers represent a class of browser extensions that modify a wide variety of sites while also providing a consistent user experience for millions of users. We discuss how the syntax framework used by these other large-scale interface-modifying tools could be applied to interface-modifying DSCTs.

2 ELEMENTS OF A SCALABLE INTERFACE-MODIFYING SYNTAX

Part of the effectiveness of password managers and ad blockers stems from the consistency of their standardized feature sets (the filling and generation of passwords, or the blocking of specific page elements, respectively) among a wide variety of websites and interface elements in the HTML domain object model. The connection between the interface elements to modify and the standardized feature set occurs through a formalized syntax: a codified array of rules which determines the behavior of these interface-modifying tools. We suggest that scalable interface-modifying syntaxes contain three components: best practices, heuristics, and quirks (Figure 1).

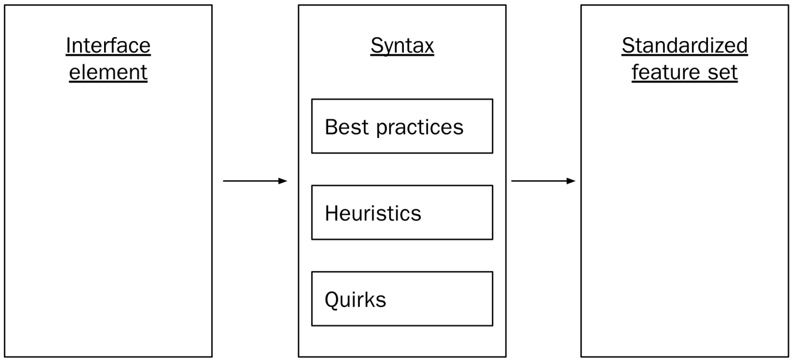

Fig. 1. A framework for scalable interface modification via the use of a syntax.

We define best practices as publicized, defined conventions that indicate how interfaces should be constructed: in the ad blocking area best practices include HTML elements that are marked by site developers with the `id` of "Adbanner" or "AD_banner" [7]. In the password manager area, best practices include recommendations from the Web Hypertext Application Technology Working Group regarding the use of the "autocomplete" attribute for passwords [8].

Next, we define heuristics as patterns which are not publicized but which provide a good indicator of a relevant design decision. In the ad blocking area, a heuristic could be that folders named "120_600" (a common ad image dimension) should be considered to be containing ads [7]. In the password manager area, a heuristic could be that "any form with at least three input elements where at least one of them is a password is an account creation form" [9].

Lastly, we define quirks as per-interface tweaks that are needed when best practices and heuristics don't cover all cases. In the ad blocking area, quirks include site-specific instructions such as "smile.fr###popup", which blocks a specific interface element on a specific site [10]. In the password manager area, quirks include site-specific instructions such as "'dowjones.com': 'password-rules': 'maxlength: 15;'", which refers to a specific password limit for a specific site [11]. Although the above quirks encapsulate elements not captured by best practices or heuristics, it may also be that some quirks are *exceptions*, or whitelists to *prevent* best practices and heuristics from capturing some elements [12].

3 DESIGNING INTERFACE-MODIFYING SYNTAXES FOR DIGITAL WELLBEING: TWO EXAMPLES

Previous literature has presented different classifications of DSCT interventions [2] [13]: within these classifications, several interventions specifically modify interfaces, including "feature minimisation," "block," "irrelevant effortful task to override," and "launch limit," among others [2]. Of this interface-modifying group, we propose two interventions to serve as example standardized feature sets for two syntaxes: (1) an irrelevant effortful task imposed when the user browses a list with infinite content/infinite scroll and (2) a reduction in color saturation for non-essential features of sites. An example of an interface element modified by (1) could be an infinite scroll of posts on a social media site; an example of an interface element modified by (2) could be the recommendations sidebar on a video streaming site.

The next step in defining this syntax would be to find if there are any *best practices* that could be used to capture interface elements: in the case of infinite content-type lists, one potential best practice could be to search for the inclusion of files from common infinite scroll libraries [14], [15]. In the case of finding non-essential webpage elements, best practices may indicate that these elements will be the first to disappear as the width of the webpage is reduced.

After this step, *heuristics* could also be defined to catch these interface elements: one heuristic used to catch infinite content-type lists could be the regular use of the *fetch* API to collect new, similarly sized blocks of information; and one heuristic to catch non-essential webpage elements could be the absence of these elements according to rules found in readability tools, such as Mozilla's Readability API [16].

Lastly, site-specific *quirks* could be added in order to make up for any failures to capture appropriate elements, such as particularly hard-to-recognize infinite content-type lists or unusual non-essential page elements. Both quirks and exceptions may be needed in order to be sure that interface modifications run smoothly.

4 IMPLICATIONS

Designing internal DSCTs with a scalable interface-modifying syntax offers several advantages over interface-specific DSCTs. The most straightforward benefit of an interface-modifying syntax is that interface-modifying DSCT interventions can adapt to new interfaces, potentially with a human in the loop. In particular, syntax-generated interventions could be improved and corrected by end-users on new interfaces through a simple proposal-based system ("Is this intervention useful?"). This proposal system would represent an improvement over current state-of-the-art DSCT intervention selection, which requires a human to be completely in control of the design of site-specific interventions as well as the determination of which generalized interventions are suitable for a website [6].

An interface-modifying syntax additionally affords the ability to classify and analyze interface elements at scale that are modified for digital wellbeing, in the same style as previous literature on dark patterns [17]. A syntax would also enable reduced manual coding in terms of DSCT design analysis and data analysis, as well as verifiable tests of intervention repeatability on unseen websites. Further, measuring a DSCT syntax at scale could comprehensively describe the complexity of the rules needed for scalable interface-modifying DSCT interventions and could illuminate commonalities in interface designs.

Several new limitations would arise with the creation of syntaxes for scalable interface-modifying DSCTs. If an interface-modifying syntax is like that of ad blockers or password managers, it must be supported by a community of users who can report bugs and update the syntax—otherwise the syntax will accumulate errors and will fail to scale up [18]. Additionally, syntaxes may become dense and repetitive, leading to reduced performance [19]. Or, aggressive interface-modifying syntaxes may be countered by new interface design strategies that attempt to avoid DSCT interventions, as previously seen in the "Ad Wars" [20]. Further, interface-modifying DSCT interventions are currently poorly suited to non-web platforms, given stronger limits on interface modification on mobile devices and other more strictly defined platforms. Yet even with these limitations, we are confident that the development of syntaxes for interface-modifying digital self-control tool interventions would present a step forward for the digital wellbeing field, and would help DSCTs reach a wider number of individuals who struggle with their digital wellbeing.

REFERENCES

[1] R.X. Schwartz. 2019. Ulysses' ropes and the inherent limits of digital self-control tools. *5th International Conference on the History and Philosophy of Computing*. DOI: 10.18130/v3-dfzq-ny16. https://doi.org/10.18130/v3-dfzq-ny16.

[2] Ulrik Lyngs, Kai Lukoff, Petr Slovak, Reuben Binns, Adam Slack, Michael Inzlicht, Max Van Kleek, and Nigel Shadbolt. 2019. Self-Control in Cyberspace: Applying Dual Systems Theory to a Review of Digital Self-Control Tools. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, Glasgow Scotland Uk, (May 2019), 1–18. ISBN: 978-1-4503-5970-2. DOI: 10.1145/3290605.3300361. Retrieved 04/11/2022 from https://dl.acm.org/doi/10.1145/3290605.3300361.

[3] Kai Lukoff, Ulrik Lyngs, Himanshu Zade, J. Vera Liao, James Choi, Kaiyue Fan, Sean A. Munson, and Alexis Hiniker. 2021. How the Design of YouTube Influences User Sense of Agency. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, Yokohama Japan, (May 2021), 1–17. ISBN: 978-1-4503-8096-6. DOI: 10.1145/3411764.3445467. Retrieved 04/19/2022 from https://dl.acm.org/doi/10.1145/3411764.3445467.

[4] Mingrui Zhang, Kai Lukoff, Raveena Rao, Amanda Baughan, and Alexis Hiniker. 2022. Monitoring Screen Time or Redesigning It?: Two Approaches to Supporting Intentional Social Media Use. In CHI Conference on Human Factors in Computing Systems (CHI '22), April 29-May 5, 2022, New Orleans, LA, *CHI Conference on Human Fac- tors in Computing Systems (CHI '22), April 29-May 5, 2022, New Orleans, LA*. DOI: 10.1145/3491102.3517722. https://doi.org/10.1145/3491102.3517722.

[5] R.X. Schwartz, Alberto Monge Roffarello, Luigi De Russis, and Panagiotis Apostolellis. 2021. Reducing Risk in Digital Self-Control Tools: Design Patterns and Prototype. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, Yokohama Japan, (May 2021), 1–7. ISBN: 978-1-4503-8095-9. DOI: 10.1145/3411763.3451843. Retrieved 05/09/2022 from https://dl.acm.org/doi/10.1145/3411763.3451843.

[6] Geza Kovacs, Zhengxuan Wu, and Michael S. Bernstein. 2021. Not Now, Ask Later: Users Weaken Their Behavior Change Regimen Over Time, But Expect To Re-Strengthen It Imminently. *arXiv:2101.11743 [cs]*, (January 2021). arXiv: 2101.11743. Retrieved 04/19/2022 from http://arxiv.org/abs/2101.11743.

[7] 2022. Easylist - easylist.txt. (2022). Retrieved 04/19/2022 from https://easylist.to/easylist/easylist.txt.

[8] 2022. HTML Living Standard - 4.10.18.7 Autofill. (2022). Retrieved 04/19/2022 from https://html.spec.whatwg.org/multipage/form-control-infrastructure.html#attr-fe-autocomplete.

[9] 2013. Chromium - password_generation_agent.cc. (2013). Retrieved 04/19/2022 from https://chromium.googlesource.com/chromium/src/+/7db5401bee32a5115268984f8461c30348346581/components/autofill/content/renderer/password_generation_agent.cc.

[10] 2022. Easylist - fanboy_annoyance_international.txt. (April 2022). Retrieved 04/19/2022 from https://github.com/easylist/easylist/blob/4598b4ad35938765f69573e00c40f3017d751bbf/fanboy-addon/fanboy_annoyance_international.txt.

[11] 2022. Apple - Password Manager Resources - password-rules.json. (2022). Retrieved 04/19/2022 from https://github.com/apple/password-manager-resources/blob/main/quirks/password-rules.json.

[12] 2022. Easylist - easylist_allowlist.txt. (April 2022). Retrieved 04/19/2022 from https://github.com/easylist/easylist/blob/4598b4ad35938765f69573e00c40f3017d751bbf/easylist/easylist_allowlist.txt.

[13] Alberto Monge Roffarello and Luigi De Russis. 2019. The Race Towards Digital Wellbeing: Issues and Opportunities. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (CHI '19). Association for Computing Machinery, New York, NY, USA, (May 2019), 1–14. ISBN: 978-1-4503-5970-2. DOI: 10.1145/3290605.3300616. Retrieved 04/11/2022 from https://doi.org/10.1145/3290605.3300616.

[14] 2022. Infinite Scroll JavaScript plugin. (2022). Retrieved 04/19/2022 from https://infinite-scroll.com/index.html.

[15] Darren Cooney. 2022. WordPressInfiniteScroll – Ajax Load More. (2022). Retrieved 04/19/2022 from https://wordpress.org/plugins/ajax-load-more/.

[16] 2022. Readability.js. (April 2022). Retrieved 04/19/2022 from https://github.com/mozilla/readability.

[17] Arunesh Mathur, Gunes Acar, Michael J. Friedman, Eli Lucherini, Jonathan Mayer, Marshini Chetty, and Arvind Narayanan. 2019. Dark Patterns at Scale: Findings from a Crawl of 11K Shopping Websites. *Proceedings of the ACM on Human-Computer Interaction*, 3, CSCW, (November 2019), 1–32. ISSN: 2573-0142. DOI: 10.1145/3359183. Retrieved 04/12/2022 from https://dl.acm.org/doi/10.1145/3359183.

[18] Aith. 2014. Blocking Based on Dimensional Filenames is Too Aggressive - Adblock Plus Forums. (2014). Retrieved 04/19/2022 from https://webcache.googleusercontent.com/search?q=cache:IGiwn2_owqsJ:https://forum.adblockplus.org/viewtopic.php%3Ft%3D25856+&cd=1&hl=en&ct=clnk&gl=us.

[19] 2018. The Mounting Cost of Stale Ad Blocking Rules. (July 2018). Retrieved 04/19/2022 from https://brave.com/the-mounting-cost-of-stale-ad-blocking-rules/.

[20] Umar Iqbal, Zubair Shafiq, and Zhiyun Qian. 2017. The ad wars: retrospective measurement and analysis of anti-adblock filter lists. In *Proceedings of the 2017 Internet Measurement Conference*. ACM, London United Kingdom, (November 2017), 171–183. ISBN: 978-1-4503-5118-8. DOI: 10.1145/3131365.3131387. Retrieved 04/19/2022 from https://dl.acm.org/doi/10.1145/3131365.3131387.