SoftwareX 18 (2022) 101090

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

WACline: A Software Product Line to harness heterogeneity in Web Annotation

Haritz Medina [*], Oscar Díaz, Xabier Garmendia

Onekin Group, Paseo de Manuel Lardizabal 1, University of the Basque Country, UPV/EHU, San Sebastian, 20018, Spain

ARTICLE INFO

Article history:
Received 19 January 2022
Accepted 13 April 2022

Keywords:
Web annotation client customization
Data curation
Browser extension
Software product lines

ABSTRACT

A significant amount of research project funding is spent creating customized annotation systems, re-inventing the wheel once and again, developing the same common features. In this paper, we present WACline, a Software Product Line to facilitate customization of browser extension Web annotation clients. WACline reduces the development effort by reusing common features (e.g., highlighting and commenting) while putting the main focus on customization. To this end, WACline provides already implemented 111 features that can be extended with new ones. In this way, researchers can reduce the development and maintenance costs of annotation clients.

Code metadata

Current code version	V0.3.1
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-22-00020
Code Ocean compute capsule	
Legal Code License	MIT
Code versioning system used	Git
Software code languages, tools, and services used	Javascript, HTML, SCSS
Compilation requirements, operating environments	Eclipse, Pure::variants v5, NodeJS v16, Gulp v4
If available Link to developer documentation/manual	https://github.com/onekin/wacline#readme
Support email for questions	haritz.medina@ehu.eus

1. Motivation and significance

Data curation is the work of organizing and managing a collection of data sets to meet the needs and interests of a specific group of people [1]. Due to the emerging growth of scientific data, scientific research increasingly relies on discovering insights from them. This led data curation to become key in research [2]. Scientific data should be available once it has been published so that other researchers can replicate the findings and conduct more experiments on it. Yet, documenting and curating data is far from trivial. It requires annotating, publishing, and presenting data available for reuse and preservation [3,4].

Annotating entails a set of activities that add more information to the data, either by identifying data structures or by adding information to contextualize aspects of the content (e.g., text labeling) [5]. Annotation practices are different for each scientific research. It is hard to prescribe a curation method that works for everyone since it must be adjusted to the specific requirements, expertise, and capability of the community at hand. Different annotation practices can be found in Social Sciences and Humanities [6], Journalism investigation [7] or Medical and Biological Sciences [8–10], just to name a few.

This scenario leads to annotation practices that share commonalities, but need to be adjusted when adopted by a given research area. As a result, researchers exert effort into developing their own annotation tools. Cohen et al. [11] reported that "in the past five years in the biomedical text mining community, we are aware of 5 projects that have developed their own tools, at an estimated cost of $500,000 US". However, usually, these efforts end as defunct projects. The *Hypothes.is* foundation conducted a survey[1] on the status of 72 annotation tools in 2014, where 58% of the annotation tools were available but most of them

[*] Corresponding author.
E-mail addresses: haritz.medina@ehu.eus (Haritz Medina), oscar.diaz@ehu.eus (Oscar Díaz), xabier.garmendiad@ehu.eus (Xabier Garmendia).

[1] https://rebrand.ly/hypothesis-survey.

https://doi.org/10.1016/j.softx.2022.101090

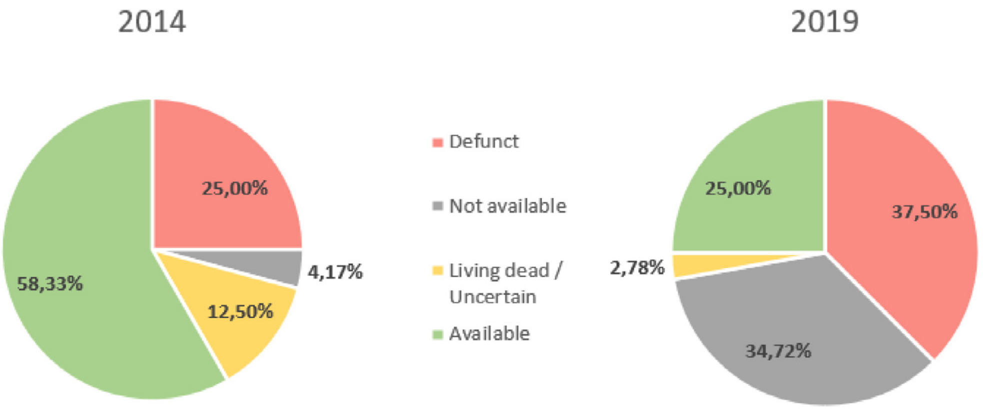


Fig. 1. Evolution of annotation projects availability: A comparison of Hypothes.is survey's annotation projects availability in 2014 and 2019.

were immature releases or their use was limited. We replicated[2] this study in 2019 to find out that only 25% of the revised tools were available, while 35% were defunct projects and 40% were no longer available (see Fig. 1).

This situation was hindered by the lack of standards. The lack of a common way to describe annotations resulted in annotation data being locked in the publishers' silos. This situation changed in 2017 with the W3C's recommendations for Web Annotation [12]. W3C provides a data model for annotation data. On these grounds, initiatives like Hypothes.is [13] aim to provide an open platform to collect Web Annotations transparently and collaboratively supporting an open backend repository for annotations. This makes consumers (rather than content providers) their annotations owners, favoring free sharing of annotations.

The practice of annotation (i.e., how annotations are produced, visualized, or shared) is very heterogeneous. W3C helps annotation portability, but it is still up to each community to build the Web annotation clients to read and write the annotation repository. Hypothes.is itself is a case in point. Hypothes.is offers a basic Web annotation client to access the annotation repository. This annotation client falls short of annotation practices in journalism, leading to bespoke annotation clients (e.g., *EJournalPress* [14] or *FakeNewsAnnotationTool* [7]). Development-wise, these initiatives follow a *Clone&Own* approach, where a new software starts by cloning an existing one, and next adapts parts of it to meet new requirements. However, although *Clone&Own* saves costs in the short run, it is not scalable if there is no way to track changes across clones. This is where *Software Product Lines* come into play.

Software Product Lines (SPLs) facilitates systematic reuse of assets in contrast with opportunistic reuse like in *Clone&Own* [15]. It aims to identify common functionality and variables between applications within a domain (e.g., Web annotation), and build reusable assets to benefit future development efforts [16]. Although the costs of implementation would be similar in both approaches, *Clone&Own* requires additional effort. It needs to filter out the code that is not going to be reused to develop the new tool, which may be time-consuming due to functionalities scattering or code coupling. In contrast, these tasks are not necessary for using an SPL approach, because here the reuse is systematic and the developer works only with the features to be reused [17].

This work presents *WACline* (Web Annotation Client-line) to face heterogeneity in annotation practices using an SPL architecture. WACline is not a one-off annotation tool but a platform for creating annotation tools. Rather than building annotation tools from scratch, WACline provides a head-start for communities to develop their own annotation tools from a set of existing components.

2. Software description

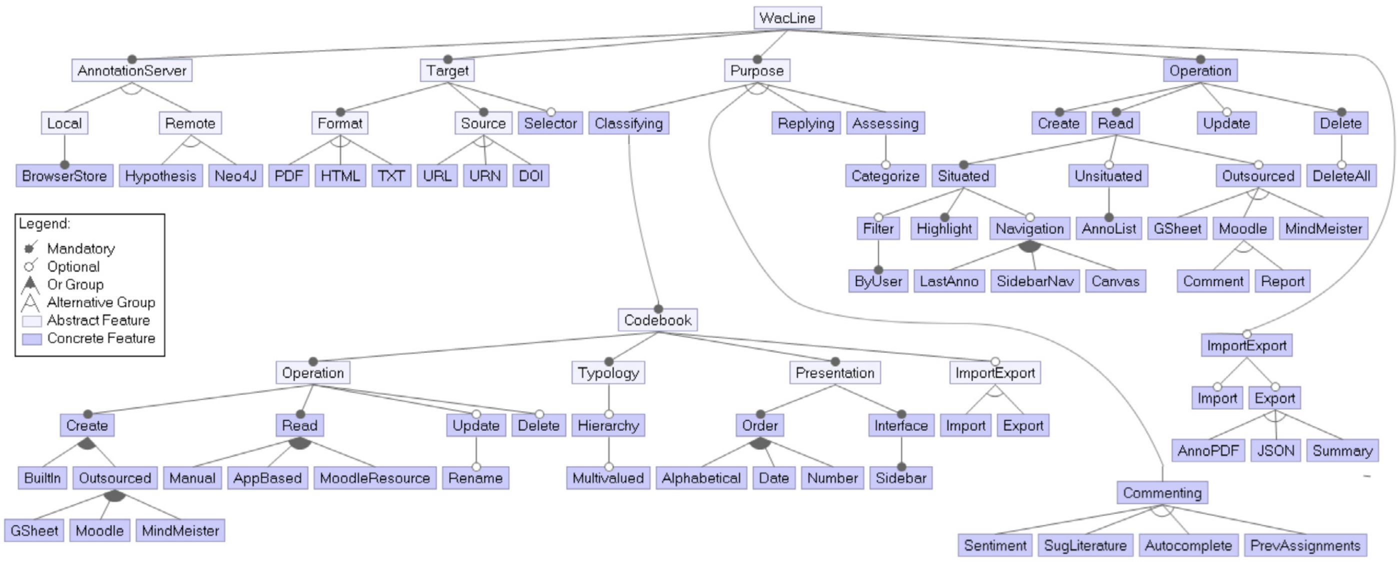
WACline offers a framework for creating custom browser extensions for Web Annotation. Specifically, browser extensions serve as the frontend (also known as annotation client) to collect and display annotations hosted in an annotation backend (also known as annotation server) (e.g., Hypothes.is). WACline provides features that can be combined in different ways to create customized annotation browser extensions to perform scientific data curation. These features are not offered in a single product. Rather, domain experts cherry-pick those features that account for the annotation practice at hand during development. To this end, *Pure::Variants* [18] is used as the variability management system.

Annotation clients are obtained from WACline along with the following steps[3]:

- **Configuration**. WACline characterizes the supported functionality in terms of a feature model (see Section 2.2). From this model, the researcher cherry-picks a set of features that better fit the requirements of the annotation practice at hand. The variability management system checks that the selected features are compliant with the existing feature dependencies (e.g., a feature to filter annotations by users requires a remote annotation server).
- **Generation**. After configuration, the instance of the new software must be created. WACline's source code is annotated with preprocessor directives, which are similar to C preprocessor directives (see Section 2.3). These preprocessor directives are used to discern commonalities (i.e., features that all Web annotation clients share) from variabilities (i.e., code implementing specific features functionality). Through a pre-processing stage, a dedicated annotation tool is generated only with the desired features, filtering out the rest of the code.
- **Customization**. Some annotation-specific requirements may not be met with WACline alone and some *ad-hoc* development is needed. This might require modifications over the source code generated by WACline to cover new functionalities.

[2] Replication of Hypothes.is survey in 2019: https://rebrand.ly/h-isSurveyReplication.

[3] Additional details are available at the User manual: github.com/onekin/WacLine/blob/master/README.md.



$$\text{Filter.ByUser} \rightarrow \text{Remote} \qquad \text{SidebarNav} \rightarrow \text{Sidebar}$$

$$\text{Canvas} \rightarrow \text{Hierarchy} \qquad \text{Replying} \rightarrow \text{Commenting}$$

Fig. 2. Partial view of WACline's Feature Diagram: structure (top) and a sample of dependencies between features (bottom).

- **Build**. Source code must be compiled to make it ready to install. To this end, we provide a script to automatically resolve dependencies and compile the resultant browser extension for the target browser (e.g., *Google Chrome*).
- **Test and Delivery**. Created output browser extension folder can be installed in the browser to test it. After testing the functionality, if it meets annotation requirements, the extension folder can be packed and released in production (e.g., published at the Chrome Web Store).

2.1. Software architecture

WACline's annotation clients follow a browser extensions architecture[4] Specifically, most of the functionality is executed in the Content Script, which augments Web pages for annotation. Content Script architecture is based on an initialization module *ContentScriptManager.js* which orchestrates the initialization of the rest of the modules: *annotationManager* (CRUD operations), *annotationStorageManager*, *codebookManager*, *targetManager*, etc.

To facilitate the inclusion of new features, WACline follows an event-driven architecture. Then, the new module's *.js* files (1) should be added to the family model (**.ccfm* file), (2) should be initialized in Content Script, and (3) should be subscribed and published to other modules' events when necessary (that are registered in *Events.js* file).

Source code is developed mainly in HTML, SCSS and Vanilla Javascript (ECMAScript 2015). NPM,[5] Webpack,[6] Babel[7] and Gulp[8] are used for dependency resolution, code transpiling, and extension building. It is published under the MIT license and it is open to receive contributions on GitHub.[9]

2.2. Software functionality

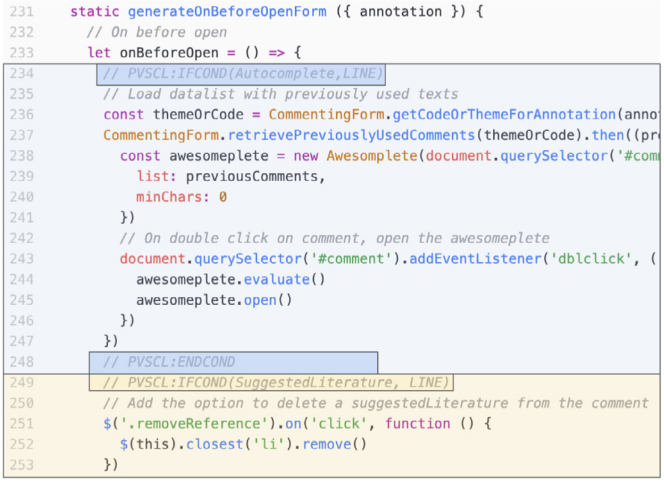
SPLs document functionality through a Feature Model [19]. It is a representation of the set of functional features supported. Fig. 2 shows the Feature Model for WACline, which is divided into six clusters that represent the main components of a Web Annotation client, namely

- *Annotation Server*, which gathers features that concern annotation storage (e.g., Hypothes.is [13]).
- *Target*, which clusters features that refer to the annotated content resource (e.g., format, source, or selector).
- *Purpose*, which refers to what annotation is created for: classifying, commenting, replying, and assessing. New purposes can be defined following W3C recommendation [20].
- *Operation*, which groups Create, Read, Update, and Delete (CRUD) operations over Web Annotations. For example, annotation reading mechanisms can be different, where different visualizations can be provided (from simple highlighting to complex tables and diagrams).
- *Codebook*, which refers to the controlled vocabulary (i.e., codes) or taxonomy (i.e., themes) used to classify annotations. This vocabulary can vary among annotation practices in their typology, presentation or how they are created and managed.
- *Import* & *Export*, which permit annotations to be imported and exported in different formats for further reuse outside the extension.

WACline's 111 features can be combined in different ways, potentially giving rise to 1.92*10^{13} full-fledged annotation tools. In addition, the code developed by researchers during the customization of annotation tools might lead to features that were not considered previously in WACline. If researchers want to contribute to the WACline project, two scenarios emerge:

- upgrade existing features, which implies modification of features' source code between preprocessor directives.

[4] https://developer.chrome.com/docs/extensions/mv3/architecture-overview/.
[5] NPMjs: https://www.npmjs.com/.
[6] Webpack: https://webpack.js.org/.
[7] BabelJS: https://babeljs.io/.
[8] Gulp: https://gulpjs.com/.
[9] Contributing: github.com/onekin/WacLine/blob/master/CONTRIBUTING.md.



```
231     static generateOnBeforeOpenForm ({ annotation }) {
232       // On before open
233       let onBeforeOpen = () => {
234         // PVSCL:IFCOND(Autocomplete,LINE)
235         // Load datalist with previously used texts
236         const themeOrCode = CommentingForm.getCodeOrThemeForAnnotation(anno
237         CommentingForm.retrievePreviouslyUsedComments(themeOrCode).then((pr
238           const awesomeplete = new Awesomplete(document.querySelector('#com
239             list: previousComments,
240             minChars: 0
241           })
242           // On double click on comment, open the awesomeplete
243           document.querySelector('#comment').addEventListener('dblclick', (
244             awesomeplete.evaluate()
245             awesomeplete.open()
246           })
247         })
248         // PVSCL:ENDCOND
249         // PVSCL:IFCOND(SuggestedLiterature, LINE)
250         // Add the option to delete a suggestedLiterature from the comment
251         $('.removeReference').on('click', function () {
252           $(this).closest('li').remove()
253         })
```

Fig. 3. *#ifdef* blocks for the *Autocomplete* and *Suggest Literature* features (partial view).

- enrich WACline extending with new features, which involves the integration of brand-new source code that can interact with other existing components of WACline via events (see Section 2.1).

2.3. Sample code snippets analysis

The following code snippet is an example of how to add new features to customize an annotation client. WACline is an annotated SPL. Annotated SPLs resort to preprocessor directives (also known as *#ifdefs*) to realize variability in code. Fig. 3 provides a snippet for the *Autocomplete* feature. If selected in the configuration step, this feature adds code for comments to be kept for reuse through an autocomplete facility. The preprocessor directive (line #234) holds the predicate: is *Autocomplete* selected? At precompile time, if *Autocomplete* is selected, the *#ifdef* block (line #235–247) is included. Otherwise, it is filtered out.

3. Illustrative examples

WACline's central premise is that most annotation practices are not entirely new, functionality-wise, but rather variants along a similar practice. This section provides two illustrative annotation tools developed out of WACline: *Highlight&Go*[10] and *Concept&Go*.[11] Both browser extensions facilitate research data curation in two different practices: development of Systematic Literature Reviews (SLR) and Concept Mapping (CM), respectively. Data curation is the process of *labeling* sources of *data* for a given *goal*. For each annotation tool, we describe the curation process, its requirements, and how are addressed by both adopting (i.e., selected features at the configuration stage) and extending (i.e., *ad-hoc* development to add new features) WACline's features.

3.1. Highlight&Go: Systematic literature reviews data extraction as an annotation practice

SLRs imply data curation where the *data* comes from the primary studies, *labeling* aims to assign codes to the text paragraphs of these studies and the *goal* is tackling a research question through literature review [21]. To reduce bias, the labeling process must be done by more than one researcher using cross-checking techniques. Results are usually reported in a spreadsheet to the scientific community [22]. In short, three main tasks are involved:

- (R1) paragraph extraction and coding,
- (R2) spreadsheet logging and,
- (R3) cross-checking and conflict resolution of coding results.

Highlight&Go adopts and extends WACline to account for these tasks [23] (see Fig. 4). First, to let researchers annotate and code papers using a configurable color-coding highlighter (R1), we adopted WACline's *Codebook* feature. Then, to transparently gather annotations in the spreadsheet (R2), WACline's *AnnotationServer* was extended with the child feature *GoogleSheet*. Finally, to facilitate cross-checking (R3), *Assessing* purpose was adopted to let supervisors assess previously made annotations by other researchers, but this feature was extended with a voting feature to validate or invalidate classification decisions. Additionally, we extended WACline with a user-based annotation filtering feature.

Highlight&Go benefited by adopting 23 WACLine features (up to 9700 SLOC), where three new features were implemented. Adoption and extension of these features took around two months by two developers part-time (a Ph.D. student and a lecturer), where most of the time was invested in the adaptation of annotation storage to *Google Sheets*.

3.2. Concept&Go: Concept mapping as an annotation practice

Concept Mapping (CM)[12] implies data curation where the *data* comes from the reading materials (e.g., research studies), *labeling* refers to assigning concepts to the text paragraphs of these materials, and the *goal* is to create a concept map of the main entities and relationships in a knowledge area. Main tasks include:

- (R1) annotate concept maps' concepts and relationships from different text resources and,
- (R2) visualize the concept map made up of the captured concepts and relationships complemented by the annotations that sustain them providing a link to the reading material to trace misconceptions.

Concept&Go adopts and extends WACline to account for these tasks (see Fig. 5). Firstly, to capture relationships (R1), we adopted *Codebook-based classification* and we extended WACline's purposes with a new purpose (*Linking*) to create links between two concepts. Secondly, to visualize the map (R2), WACline was extended with *CXLExport* feature to export the gathered concepts and relationships with the annotations to CmapTools.

Concept&Go benefited from 17 WACLine features (up to 10250 SLOC). Only two new features were necessary to support the creation of relationships and concept map visualization, though they account for 3160 SLOCs. Adoption and extension of these features took around four months.

4. Impact

In the last five years, up to seven different surveys were published [25–31] which analyze more than 200 annotation tools used in linguistics, education, biology, and e-health research. This denotes the high number of investigations that can be potentially benefited from the use of WACline.

[10] https://rebrand.ly/highlightAndGo.

[11] https://rebrand.ly/conceptAndGo.

[12] Concept Mapping (CM) is the act of reflecting the organization and understanding of knowledge in a diagram made up of concepts and relationships among them. They help researchers describe their structure of knowledge in communication, and they promote knowledge sharing and the creation of new knowledge [24].

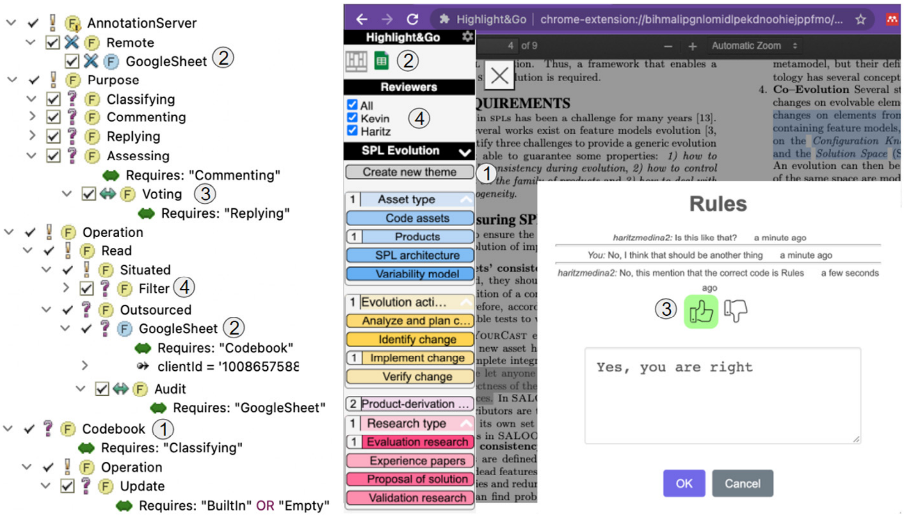


Fig. 4. Highlight&Go annotation client configuration using Pure::Variants notation (partial view). (1) A dedicated codebook to classify annotations. (2) Storage of annotations in *Google Sheets*. (3) Voting to cross-check (validate/invalidate) researchers' classification. (4) User-based filtering mechanism.

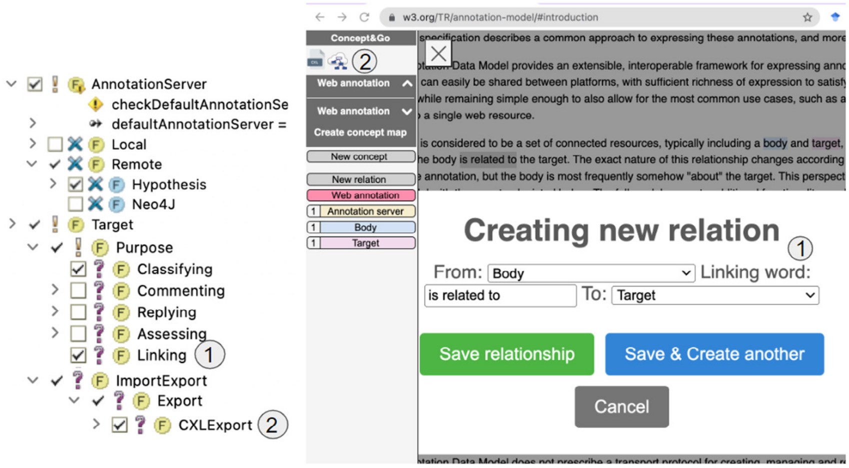


Fig. 5. Concept&Go variant definition using Pure::Variants notation (partial view). *Linking* feature creates a button in the sidebar to allow users to relate two concepts using a linking word, in the example "Body" and "Target" concepts are linked using "is related to" linking word (1). Buttons to export a Concept Map in CmapCloud using CXL format (2).

4.1. Development and maintenance cost

WACline represents a step towards annotation tooling in the scientific discovery process. By adopting and extending WACline's features, researchers can support their annotation-based investigation processes while reducing their development costs. Annotation client development cost is reduced by reusing common annotation functionalities in WACline's core assets but also reusing the already implemented 111 configurable features.

Furthermore, if the creation of the annotation tool has led to the development of new functionalities, researchers can incorporate those novelties into WACline in terms of new features. In order to integrate feature modifications and the implementation of new requirements, the annotation tool developer must annotate the new code with the preprocessor directives as explained in 2.3. These extensions can be shared among the community to create variants that can support new annotation practices, combining some of these new features and further extensions.

In the same way, feature maintenance (e.g., bug fixing) and evolution benefit not only the annotation client updated, but also others which use the same feature too. This makes fixes and changes need to apply only once, which may increase the quality of developed Web Annotation tools.

The bottom line is that as development and maintenance costs are reduced, research projects funding can be better invested in scientific discovery.

4.2. Interoperability

The majority of annotation tools do not follow any standard to represent annotations' data model [27]. This hinders important aspects of research, such as reproducibility or reuse of annotation data sets (e.g., requiring transformation between annotation tools' data models) [32]. This problem has been solved by the W3C annotation data model [12]. However, there is still a gap between W3C recommendations and real practice where even new annotation tools keep using custom formats.

WACline follows W3C's data annotation model. Annotations generated through WACline tools can then be consumed by other tools that also follow the W3C recommendation, solving existing interoperability issues in the area [33].

5. Conclusions

We have presented WACline, an SPL aimed at bringing systematic reusability to the Web annotation client world. WACline offers 111 features research communities can configure to fit their annotation practices (for annotation creation, consumption, and manipulation). This prevents communities from reinventing the wheel while keeping required customization for annotation practices across different areas, such as biomedical, linguistics, or jurisprudence. We have shown the feasibility of developing annotation tools from WACline with two working examples that are available in the Chrome Web Store.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Thanks are due to Jeremías P. Contell, Iker Azpeitia, Aitor Díaz de Otazu, Gorka Arce and Iñigo Bereciartua who participate in the WACline initiative implementing their own annotation tools.

Funding

Open Access funding was provided thanks to the CRUE-CSIC agreement with Elsevier. This work was supported by the Spanish Ministry of Science, Innovation and Universities under contract RTI2018-099818-B-I00 and TIN2017-90644-REDT. Onekin Group enjoys support from the program Grupos de Investigación del Sistema Univesitario Vasco 2019–2021 under contract IT1235-19. Haritz Medina and Xabier Garmendia enjoy a grant from the University of the Basque Country - PIF17/15 and PIF20/236 respectively.

References

[1] Cavanillas JM, Curry E, Wahlster W. New horizons for a data-driven economy: a roadmap for usage and exploitation of big data in Europe. Springer Nature; 2016.

[2] Ou S, Zhou Y. Current status of scientific data curation research and practices in Mainland China. LIBRES: Libr Inf Sci Res Electron J 2016;26(1).

[3] Agosti M, Di Nunzio GM, Ferro N. The importance of scientific data curation for evaluation campaigns. In: International DELOS conference. Springer; 2007, p. 157–66.

[4] Gray J, Szalay AS, Thakar AR, Stoughton C, VandenBerg J. Online scientific data curation, publication, and archiving. In: Virtual observatories, vol. 4846. International Society for Optics and Photonics; 2002, p. 103–7.

[5] Ajay C, Fabio C, Vitakeska L. Cross-media document annotation and enrichment. 209, 2006, p. 9P,

[6] Caria F, Mathiak B. Annotation in digital humanities. In: Digital cultural heritage. Springer; 2020, p. 39–50. http://dx.doi.org/10.1007/978-3-030-15200-0_3.

[7] Rehm G, Moreno-Schneider J, Bourgonje P. Automatic and manual web annotations in an infrastructure to handle fake news and other online media phenomena. In: LREC 2018 - 11th international conference on language resources and evaluation. 2018, p. 2416–22, https://aclanthology.org/L18-1384.

[8] Cejuela JM, McQuilton P, Ponting L, Marygold SJ, Stefancsik R, Millburn GH, et al. Tagtog: Interactive and text-mining-assisted annotation of gene mentions in PLOS full-text articles. Database 2014;2014:1–8. http://dx.doi.org/10.1093/database/bau033.

[9] Zhang W, Ning Y, Yu Zhang L, Chou E, Boyce R, Sibilla M. AnnotationPress. 2017, URL https://github.com/dbmi-pitt/dbmi-annotator.

[10] Rehm HL, Berg JS, Brooks LD, Bustamante CD, Evans JP, Landrum MJ, et al. Clingen—the clinical genome resource. N Engl J Med 2015;372(23):2235–42. http://dx.doi.org/10.1056/NEJMsr1406261.

[11] Cohen KB, Demner-Fushman D, Fort K, Grouin C, Hunter LE, Leser U, et al. Towards the last annotation tool. Tech. rep., linkedannotation; 2016, URL http://blahmuc.linkedannotation.org/resources/applications/BLAHmuc2016_paper_1.pdf.

[12] Sanderson R, Ciccarese P, Young B. W3C web annotation recommendation. In: W3C. 2017, URL https://www.w3.org/annotation/.

[13] Hypothesis. Hypothes.is project. 2022, URL https://web.hypothes.is/.

[14] EJournalPress, Hypothesis project and American Geophysical Union. AGU Launches hypothesis to facilitate peer review, hypothes.is. 2017, URL https://web.hypothes.is/blog/agu-launches-hypothesis-to-facilitate-peer-review/.

[15] Clements P. Control channel toolkit : A software product line case study. Tech. Rep. September, Defense Technical Information Center; 2001, URL https://apps.dtic.mil/docs/citations/ADA396286.

[16] Apel S, Batory D, Kästner C, Saake G. Feature-oriented software product lines. Springer; 2013, http://dx.doi.org/10.1007/978-3-642-37521-7.

[17] Echeverría J, Pérez F, Panach JI, Cetina C. An empirical study of performance using clone & own and software product lines in an industrial context. Inf Softw Technol 2021;130:106444. http://dx.doi.org/10.1016/j.infsof.2020.106444.

[18] Beuche D. Industrial variant management with pure::variants. In: ACM international conference proceeding series, vol. B. New York, New York, USA: Association for Computing Machinery; 2019, p. 1–3. http://dx.doi.org/10.1145/3307630.3342391.

[19] Lee K, Kang KC, Lee J. Concepts and guidelines of feature modeling for product line software engineering. In: Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics). 2319, 2002, p. 62–77. http://dx.doi.org/10.1007/3-540-46020-9_5.

[20] Sanderson R, Ciccarese P, Young B. W3C annotation model - motivation and purpose, world wide web. 2017, URL https://www.w3.org/TR/annotation-model/#motivation-and-purpose.

[21] Kitchenham B, Budgen D, Brereton OP. Evidence-based software engineering and systematic reviews. ACM; 2015, URL https://dl.acm.org/doi/book/10.5555/2994449.

[22] Tell P, Cholewa JB, Nellemann P, Kuhrmann M. Beyond the spreadsheet: Reflections on tool support for literature studies. In: Proceedings of the 20th international conference on evaluation and assessment in software engineering. 2016, http://dx.doi.org/10.1145/2915970.2916011.

[23] Díaz O, Medina H, Anfurrutia FI. Coding-data portability in systematic literature reviews. In: Proceedings of the evaluation and assessment on software engineering. New York, New York, USA: Association for Computing Machinery (ACM); 2019, p. 178–87. http://dx.doi.org/10.1145/3319008.3319025.

[24] Wang X-Y, Mu Y. Visualization based on concept maps: An efficient way to knowledge sharing and knowledge discovery in e-science environment. In: 2009 Sixth international conference on fuzzy systems and knowledge discovery, vol. 2. IEEE; 2009, p. 144–7. http://dx.doi.org/10.1109/FSKD.2009.540.

[25] Neves M, Ševa J. An extensive review of tools for manual annotation of documents. Brief Bioinform 2019. http://dx.doi.org/10.1093/bib/bbz130.

[26] Ghadirian H, Salehi K, Ayub AFM. Social annotation tools in higher education: a preliminary systematic review. Int J Learn Technol 2018;13(2):130–62. http://dx.doi.org/10.1504/IJLT.2018.092096.

[27] Kalboussi A, Mazhoud O, Kacem AH. Comparative study of web annotation systems used by learners to enhance educational practices: Features and services. Int J Technol Enhanc Learn 2016;8(2):129–50. http://dx.doi.org/10.1504/IJTEL.2016.078081.

[28] Gayoso-Cabada J, Sarasa-Cabezuelo A, Sierra J-L, Scho A, Sierra J-L. Document annotation tools: Annotation classification mechanisms. In: Proceedings of the sixth international conference on technological ecosystems for enhancing multiculturality. 7, ACM; 2018, p. 889–95. http://dx.doi.org/10.1145/3284179.3284331.

[29] Breitenfeld A, Müller-Birn C. A state-of-the-art review of semantic annotation tools. Tech. rep., Freie Universität; 2017, URL https://refubium.fu-berlin.de/bitstream/handle/fub188/21870/technical-report-annotation-tools.pdf.

[30] Krouska A, Troussas C, Virvou M. Social annotation tools in digital learning: A literature review. In: 2018 9th International Conference on Information, Intelligence, Systems and Applications. IEEE; 2018, p. 1–4. http://dx.doi.org/10.1109/IISA.2018.8633609.

[31] Chehab K, Kalboussi A, Kacem AH. Study of healthcare annotation systems. Int J E-Health Med Commun (IJEHMC) 2021;12(3):74–89. http://dx.doi.org/10.4018/IJEHMC.20210501.oa5.

[32] Ambrósio LM, Marques P, David JMN, Braga R, Ribeiro Dantas MA, Ströele V, et al. An approach to support data integration in a scientific software ecosystem platform. In: 2019 IEEE 23rd international conference on computer supported cooperative work in design. 2019, p. 39–44. http://dx.doi.org/10.1109/CSCWD.2019.8791499.

[33] Ciccarese P, Soiland-Reyes S, Clark T. Web annotation as a first-class object. IEEE Internet Computing 2013;17(6):71–5. http://dx.doi.org/10.1109/MIC.2013.123.