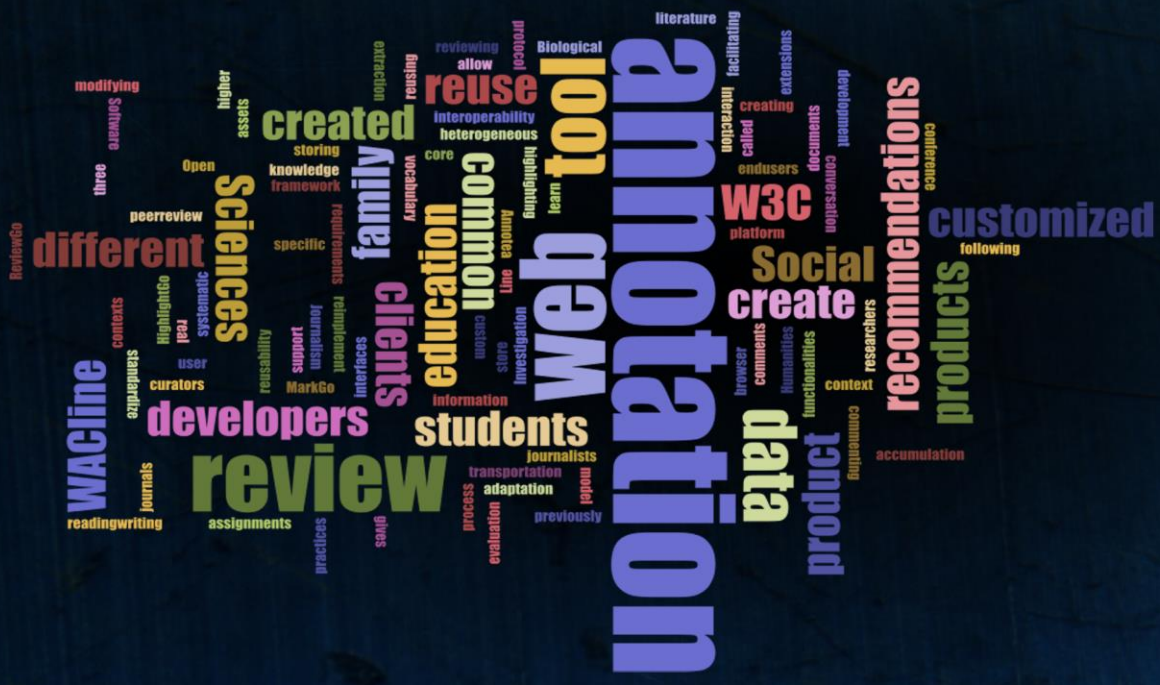


Ph.D. Thesis

Harnessing customization in Web Annotation: A Software Product Line approach

Haritz Medina Camacho

2022

eman ta zabal zazu

Universidad del País Vasco

Euskal Herriko Unibertsitatea

Supervisors: Oscar Díaz
Maider Azanza

Harnessing customization in Web Annotation: A Software Product Line approach

Dissertation
presented to
the Department of Computer Languages and Systems of
the University of the Basque Country
in Partial Fulfillment of
the Requirements for the Degree of

Doctor of Philosophy
("*international*" mention)

Haritz Medina Camacho

Supervisors:
Prof. Dr. Oscar Díaz García
Dr. Maider Azanza Sesé
San Sebastián, Spain, 2022

This work was hosted by the *University of the Basque Country* (Faculty of Informatics). The author enjoyed a doctoral grant from the University of the Basque Country under the contract PIF17/15 from 2018 to 2022. The work was co-supported by the *Spanish Ministry of Science, Innovation, and Universities*, the *Spanish State Research Agency*, and the *European Regional Development Fund* under contract RTI2018-099818-B-I00 and TIN2017-90644-REDT.

Resumen

Las anotaciones permiten proporcionar información extra asociada a un fragmento en particular de un documento o pieza de información. Esas notas se suelen utilizar para dar comentarios, inspirar el debate o facilitar el proceso de aprendizaje. Las anotaciones se llevan utilizando durante siglos en formato físico, como puede ser en papel o en libros. A medida que toda la información se está digitalizando, y más en concreto, trasladando a la web, surge la necesidad de realizar anotaciones también en la web. Desde la creación de la web, Tim Berners-Lee concibió las anotaciones como una capa sobre la web donde los usuarios podían complementar la web con sus propias notas. Desde la creación de la primera herramienta de anotación en 1994 se han creado cientos de aplicaciones, desde herramientas de propósito general (como *Hypothes.is* o *Diigo*) hasta herramientas de anotación especializadas para abordar tareas más específicas dentro de contextos como la biomedicina, educación o ciencias sociales.

Anotaciones interoperables: W3C Web Annotation recommendation

Debido al aumento del número de herramientas de anotación, en los últimos años, se ha tratado de buscar una manera de estandarizar las anotaciones en la web. Se han realizado diversos intentos como Annotea, Open Annotation y Annotation Ontology, hasta que finalmente, fruto de todo esfuerzo previo, el W3C definió las recomendaciones de anotación web en 2017. Estas recomendaciones tienen como objetivo facilitar la interoperabilidad de las anotaciones web. Para ello, el W3C ha definido el modelo de datos de las anotaciones (compuesto principalmente por el Body, qué información extra se añade, y el Target, en qué documento o punto de información se añade), el vocabulario de ese modelo de datos (ontología) y el protocolo de transporte (comunicación entre clientes de anotación y servidores de anotación). Estas recomendaciones describen cómo se deben definir o transportar las anotaciones, pero deja sin restricciones el proceso de anotación que permita a los usuarios llegar a sus objetivos. Estos objetivos pueden ser diversos y para facilitar la descripción de los mismos, el W3C introduce las motivaciones (con qué objetivo o para qué se realiza la anotación) y

propósitos (con qué objetivo o para qué se añade esa información extra). Por ejemplo, la motivación por la cual se puede crear una anotación es para usarlo a modo de marcador (*oa:motivation oa:bookmark*) un texto interesante que luego quiera recuperar o el propósito por el cual añades un texto sea para describir (*oa:purpose oa:describing*) ese marcador. El cómo se crean las anotaciones, en gran medida depende de los propósitos que tengan los usuarios. Por ejemplo, para hacer un comentario, el usuario necesita una interfaz en la cual pueda escribir un comentario, un campo de texto. Por otro lado, si el usuario necesita clasificar un fragmento de texto, necesitará poder definir y utilizar una taxonomía. Es en este punto donde surge la gran cantidad de herramientas de anotación específicas.

Heterogeneidad de las herramientas de anotación y su coste de desarrollo

Como hemos comentado antes, las herramientas de anotación se utilizan en múltiples actividades como pueden ser: la anotación de genes en artículos de biología, procesos de aprendizaje mediante la lectura en educación o verificación de hechos en periodismo. Es por ello que una herramienta no puede abordar todas las prácticas de anotación existentes. En lo que a arquitectura se refiere, las herramientas de anotación digitales se dividen en 3 arquitecturas: aplicaciones de escritorio, sitios web que permiten la anotación y extensiones de navegador. Sin embargo, dependiendo de la práctica los usuarios pueden tener la necesidad de anotar contenido en ficheros locales (como un PDF) o en la web (páginas HTML o videos, entre otros). Es por ello que las herramientas más populares son las extensiones de navegador, ya que permiten ambas posibilidades. Entre ellas se encuentran extensiones como *Hypothes.is*, *Diigo* o *Kami*, que son de propósito general. Sin embargo, como hemos comentado antes, las herramientas de anotación de propósito general no se adaptan siempre a las prácticas que necesitan los usuarios. Es por ello que algunos proyectos adaptan estas herramientas a sus propios contextos como es el caso de *EJournalPress*, donde se adapta *Hypothes.is* para soportar la revisión por pares o *FakeNewsAnnotation-Tool* donde se adapta para soportar la verificación de noticias. Aun así, este método dificulta la mantenibilidad, por ejemplo, si *Hypothes.is* cambia ciertos aspectos de su herramienta, estos han de propagarse a estas herramientas. Lo cual, no es fácil y no siempre se puede hacer. En consecuencia, muchos desarrolladores deciden desarrollar desde cero esas herramientas a pesar de tener que volver a re-implementar funcionalidades que ya estaban implementadas en otras herramientas. Analizando el coste, tanto de adaptar como de desarrollar desde cero, es alto (cercano a 18 meses de desarrollo de media de los proyectos que hemos analizado). De igual manera, y en base a un estudio de *Hypothes.is* que hemos replicado, más de la mitad de las extensiones que estaban mantenidas en 2014 en 2019 ya estaban desmantenidas o abandonadas por sus desarrolladores.

Metodología de investigación: Action design research

Metodológicamente, esta tesis está abordada siguiendo la metodología de Action Design Research (ADR). ADR es la combinación de Design Science Research (DSR) y Action Research (AR). AR tiene como objetivo contribuir tanto a las preocupaciones prácticas de las personas en un contexto concreto, como a los objetivos científicos mediante la colaboración con los profesionales del contexto. DSR, a su vez, no se conforma con describir, explicar o predecir el mundo, si no también pretende cambiarlo o mejorarlo. Para ello, se desarrollan artefactos que solucionan problemas del mundo real. La combinación de ambos da como resultado ADR, cuyo objetivo es generar conocimiento de diseño prescriptivo a través de la construcción y evaluación de artefactos en un entorno organizacional de manera iterativa (o en ciclos). Es por ello que para cada uno de los problemas abordados dentro de la tesis se realizan varias iteraciones involucrando a usuarios reales tanto para el diseño como para la evaluación de los artefactos. Entre las tareas está la definición del problema (donde en este caso se enmarca dentro de un contexto concreto y luego se generaliza), definición de los principios de diseño que resuelven el problema (que se llevan a cabo basándose en la práctica de los profesionales del contexto), la implementación de esos principios en un artefacto y su posterior evaluación.

Problemas abordados en la tesis

Partiendo de las premisas de que las prácticas de anotación son heterogéneas y que el coste de desarrollo y mantenimiento es alto, la propuesta dentro de esta tesis es la de ofrecer una plataforma que facilite el desarrollo de extensiones de anotación web para revisión de documentos soportando la reutilización sistemática. Para ello se han desarrollado tres herramientas de anotación que sirven para mejorar la práctica en tres contextos de revisión específicos mediante el uso de la anotación. En concreto:

- Utilización de una herramienta de anotación específica para mejorar la eficiencia y efectividad del proceso de extracción de datos en revisiones sistemáticas de la literatura.
- Utilización de una herramienta de anotación específica para mejorar la calidad del feedback en la revisión y corrección de tareas de alumnos universitarios en un contexto de evaluación continua
- Utilización de una herramienta de anotación específica para mejorar la calidad de las revisiones por pares de artículos académicos

Sin embargo, en lugar de desarrollar tres herramientas desde cero, estas siguen un proceso de acumulación de conocimiento siguiendo un proceso sistemático que permite no sólo la reutilización del conocimiento generado en cada

proyecto, si no del diseño de los artefactos desarrollados y su implementación. El resultado de este proceso es una línea de producto software que permite la reutilización de manera sistemática para reducir el coste de desarrollo y mantenimiento de las herramientas de anotación dentro del dominio de revisión de documentos.

A continuación, vamos a profundizar en cada una de las problemáticas abordadas mediante customización de herramientas de anotación y posteriormente describir el proceso de acumulación de conocimiento dentro de proyectos DSR con el cual hemos ido desarrollando la línea de producto software resultante que permite la customización de herramientas de anotación para revisión de documentos.

Extracción de datos en revisiones sistemáticas de la literatura

Las revisiones sistemáticas de la literatura (SLR) y los mapeos sistemáticos de la literatura (SMS) implica recopilar y analizar datos de estudios primarios para responder preguntas de investigación en un campo o área determinada. Una de las etapas más exigentes es la extracción de datos. El objetivo de esta etapa es extraer datos de estudios primarios para luego abordar las preguntas de investigación de la revisión de la literatura. Tradicionalmente, la extracción de datos se realiza utilizando un formulario de extracción (un formulario, en papel o web donde se extraen datos o evidencias para contestar esas preguntas de investigación). En este proceso, no es raro que esa codificación se haga en papel o de manera digital mediante un visor PDF y luego trasladar los resultados a una hoja de cálculo. Aunque existen herramientas más sofisticadas como nVivo, la curva de aprendizaje es alta y los datos no son portables para compartirlos y reutilizarlos por los propios investigadores o por terceros.

Esto explica por qué la hoja de cálculo es la herramienta predominante en la extracción de datos. Son fáciles de utilizar y de compartir. Sin embargo, en ella los autores simplemente plasman los resultados finales, pero no el proceso seguido, con lo que se dificulta la validación de los resultados obtenidos. Normalmente las evidencias quedan anotadas en los documentos PDF y los resultados de clasificación en la hoja de cálculo, dificultando la trazabilidad, tanto durante el proceso como a posteriori. Gracias a que el W3C define las anotaciones como recursos web, estos pueden ser referenciables mediante la hoja de cálculo, facilitando esa trazabilidad y haciendo la extracción más eficaz. Sin embargo, pedir a los extractores que vayan creando esas anotaciones y referenciarlas en la hoja de cálculos es laborioso.

Con el objetivo de abordar este problema, se plantea el uso de una herramienta de anotación que permita la clasificación de estudios primarios y la generación de la hoja de cálculo de manera automática en base a las anotaciones que realizan los extractores. El resultado es *Highlight&Go*, una herramienta de anotación basada en la codificación por colores que genera una hoja de cálculo con la clasificación realizada por los extractores en una SLR o SMS y las referencias a las evidencias anotadas en la versión web o documentos PDF de los

estudios primarios. De igual manera, soporta la extracción independiente y la detección automática de conflictos y resolución de conflictos entre extractores. Para la evaluación de la solución se ha realizado una evaluación cualitativa y cuantitativa. Para la cualitativa tres investigadores han realizado la parte de extracción de datos de su SLR o SMS respectiva mediante el uso de *Highlight&Go*. Se ha realizado un seguimiento mediante los diary studies durante los meses que lo han utilizado y un focus group confirmatorio al final de cara a validar la utilidad de la herramienta para esta práctica.

Corrección de tareas en un contexto de evaluación continua

En la educación universitaria existe una tendencia cada vez mayor para trasladar la evaluación a una evaluación más continua, donde se les realiza un seguimiento más continuado a los estudiantes y permite actuar en consecuencia para mejorar su proceso de aprendizaje. Aquí, la retroalimentación se vuelve fundamental, donde el objetivo no es solamente calificar sino obtener información sobre el progreso de los estudiantes. La retroalimentación o feedback, en este contexto, se convierte en una piedra angular para que los estudiantes puedan ver su progreso y mejorar de manera significativa su proceso de aprendizaje. Nicol en 2010 proporcionó una serie de características que un feedback de calidad ha de cumplir, entre los que destaca que este debe ser personal (referido a lo que el estudiante conoce y no), contextual (enmarcado dentro de los criterios de evaluación), específico (haciendo referencias a la tarea realizada por el alumno) y este feedback a su vez se debe proporcionar a tiempo (es decir, que permita al alumno mejorar antes de la siguiente entrega). El problema aquí no es sólo brindar feedback de calidad, si no ver cómo entregar ese feedback a tiempo a gran escala, es decir, en cursos universitarios donde participan muchos alumnos. En la literatura esto se ha definido como una triple restricción entre tiempo, calidad y número de estudiantes.

Con el objetivo de abordar este problema, se plantea el uso de una herramienta de anotación que permita la corrección de las tareas que los alumnos suben a un sistema para la gestión del aprendizaje (como puede ser Moodle). El resultado es *Mark&Go*, una herramienta de anotación que trabaja sobre ejercicios en Moodle. Mediante el uso de la anotación, se puede anotar las tareas que los alumnos suban en formato digital (proveyendo feedback especifico) ligarlo mediante anotaciones de color a una rubrica de evaluación (proveyendo feedback contextualizado), permitiendo relacionarlo con tareas anteriores y reutilización de comentarios (proveyendo un feedback personalizado) y automatizando la publicación de los resultados en Moodle (mejorando el tiempo en el que se da feedback de calidad). Para la evaluación de la solución se ha realizado una evaluación cualitativa donde cuatro profesores han utilizado la herramienta en varios cursos para proveer feedback a sus alumnos y se ha realizado un focus group confirmatorio final de cara a validar la utilidad de la herramienta para esta práctica.

Revisión por pares de artículos científicos

La revisión por pares de los artículos está bajo presión. A pesar de ser un aspecto esencial para regular el sistema de publicación académica, esta es realizada por revisores que realizan esta labor de manera altruista y que, a su vez, son personas muy ocupadas. Los autores valoran muy positivamente el feedback que reciben, pero se quejan de que es un proceso lento hasta el punto de poner en juego la investigación. Aunque parte de este proceso de revisión se haya trasladado a la web, donde la comunicación entre revisores, editores y autores se realiza en su totalidad de forma digital, el proceso de revisión se sigue realizando de manera manual, en papel o mediante un visor de PDF. Iniciativas como *EJournalPress* promovida por el American Geophisical Union y Hypothes.is permiten la revisión por pares de manera colaborativa. Sin embargo, se centran más en la parte colaborativa, que es importante, pero la revisión no es únicamente anotar y comentar. La revisión por pares requiere de una gran exigencia, sobre todo para realizar una buena revisión, y más si cabe teniendo en cuenta las agendas tan apretadas de los investigadores más experimentados. Es por ello que existe cierta disyuntiva entre ofrecer una retroalimentación a los autores de los manuscritos de calidad y en un tiempo corto o a un coste bajo. Con el objetivo de abordar este problema, se plantea el uso de una herramienta de anotación que permita orientar a los revisores para reducir el esfuerzo que requieren para la revisión de artículos de investigación. El resultado es *Review&Go*, una herramienta de anotación específica basada en anotación en colores en base a un marco de revisión. Para ello se han reformulado los aspectos de buen feedback en educación, adaptandolos al contexto de la revisión por pares, donde se busca una retroalimentación contextualizada (pero en base al marco de revisión), específica (referenciando al manuscrito), selectiva (que se comenten los únicamente los aspectos más destacables positiva y negativos del manuscrito) y a su vez que se proporcione a tiempo. Para ello como resultado de la anotación se genera un borrador que sirve como punto de partida del informe de revisión que se debe escribir a los autores del artículo. Para la evaluación de la solución se ha hecho una evaluación preliminar donde un grupo de investigadores con experiencia en revisiones han probado en una sesión de revisión las funcionalidades principales de *Review&Go* para proveer feedback y han puntuado la utilidad percibida y la facilidad de uso para la misma.

Acopio de conocimiento del diseño a través de la reutilización sistemática para desarrollar una línea de producto software de extensiones de anotación

Dentro del contexto de DSR los proyectos no son estancos donde un proyecto aborda un problema, diseña una solución y la evalua. Los resultados de ese proyecto DSR sirven para nutrir la base de conocimiento de cara a que otros investigadores puedan reutilizar el conocimiento para construir sobre él. Durante el desarrollo de estas herramientas de anotación, se ha seguido también un proceso de acopio de conocimiento del diseño, donde el resultado del proyecto

Highlight&Go es el punto de partida para *Mark&Go* y dónde el resultado de *Mark&Go* ha servido para informar al proyecto *Review&Go*. En la literatura de DSR se ha trabajado sobre el cómo reutilizar el conocimiento, pero en menor medida el cómo reutilizar el software de investigación. Para soportar esta reutilización, acudimos a las líneas de producto software donde hemos adaptado el proceso de DSR añadiendo un ciclo adicional, el de fitness. De esta manera preparamos el software desarrollado en un proyecto, como puede ser *Highlight&Go* a una plataforma que permita su reutilización sistemática, como es una SPL. En este caso, se realizan anotaciones sobre el código fuente del artefacto identificando los mecanismos implementados, de tal manera que el que quiera reutilizar ese tenga la posibilidad de crear configuraciones alternativas que permitan la adaptación a su contexto del problema y al diseño de la solución que se requiere. Este proceso lo hemos instanciado a lo largo de la tesis, obteniendo como resultado una SPL de anotación llamada *WACline*, que ha sido refactorizada de cara a poder soportar otros contextos de anotación más allá de los inicialmente previstos en esta tesis.

Para la evaluación de la solución por un lado se ha medido la reutilización y el mantenimiento de las herramientas de anotación desarrolladas, donde se refleja una reducción de los costes gracias al uso de una línea de producto software. Por otro lado, gracias a terceros desarrolladores se han desarrollado, a partir de *WACline*, otras 3 herramientas de anotación para la creación de mapas mentales en procesos de aprendizaje en un contexto de educación, corrección de trabajos de fin de grado y revisión de sentencias jurídicas. De esta manera, se ha probado el coste de desarrollo de estas herramientas personalizadas que ha sido en promedio de unos 4 meses por un desarrollador único y también para probar la heterogeneidad y fácil extensibilidad de las funcionalidades soportadas por la línea de productos creada.

Summary

Web annotation is a common and social behavior that helps to mediate reading-writing interaction by conveying information, adding comments, and inspiring conversation in web documents. It is used in areas from Social Sciences and Humanities, Journalism Investigation, Biological Sciences or Education, just to name a few. Annotation activities are heterogeneous, where end-users (students, journalists, data curators, researchers, and so on) have very different requirements for creating, modifying, and reusing annotations. This resulted in a large amount of web annotation tools and different ways to represent and store web annotations. To facilitate reuse and interoperability, several attempts have been made during the last decades to standardize web annotations (e.g., Annotea or Open Annotation) resulting in the W3C Annotation recommendations published in 2017. W3C recommendations provide a framework for annotation representation (data model and vocabulary) and transportation (protocol). However, there is still a gap in how annotation clients (tools and user interfaces) are developed, making developers reimplement common functionalities (i.e., highlighting, commenting, storing,...) to create their customized annotation tool.

This thesis aims to provide a reuse platform for the development of web annotation tools for review. To this end, we operationalize this vision through a Software Product Line called *WACline*. *WACline* is a family of annotation products that allow developers to create custom web annotation browser extensions, facilitating the reusability of core assets and their adaptation to their specific review context. It was created following a knowledge accumulation process where each annotation product learns from previously created annotation products. Finally, we reach a family of annotation clients that gives support for three reviewing practices: systematic literature review data extraction (*Highlight&Go*), students' assignments review in higher education (*Mark&Go*), and conference and journals peer-review (*Review&Go*). For each of the review contexts, an evaluation with real stakeholders has been conducted to validate efficiency and effectiveness improvements brought by customized annotation tools in their practice.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

This chapter introduces the thesis and the motivation behind this work. Section 1.2 contextualizes the research. Section 1.3 introduces the problem addressed in this thesis and Section 1.4 presents to what extent the research question (RQ) is relevant. Next, Section 1.5 describes the research approach and methodology used in this dissertation. Finally, Section 1.6 outlines the contents of the remaining chapters.

1.2 Context

This work is framed in the area of web annotation, i.e., annotation conducted on the web. Annotations are a type of *marginalia*, where the notes in the margin are associated with a particular point in the document. Traditionally, annotations have been made on paper or in books. With the advent of digitization and the web, documents were moved to the web making them available online. Annotation becomes web annotation when the setting in which annotation is performed is the web. This changes not only how annotations are created but also increases the possibility of sharing or co-working. Since their adoption, hundreds of web annotation tools have been developed, for general use or specialized fields.

To facilitate interoperability and standardization among annotation tools, in 2017 the W3C released the recommendations for web annotations [W3C17]. W3C standardizes the annotation-as-a-noun, i.e., how annotations should be described. However, it leaves the annotation-as-a-verb unconstrained, i.e., the annotation process in the pursuit of user goals, which might be diverse. This heterogeneity is supported by motivation and purpose in W3C. Motivation and purpose describe the why and what a user creates an annotation for. For example, a web annotation can be created to bookmark a fragment of text on a website or place a textual comment aside, adding a new idea developed by the

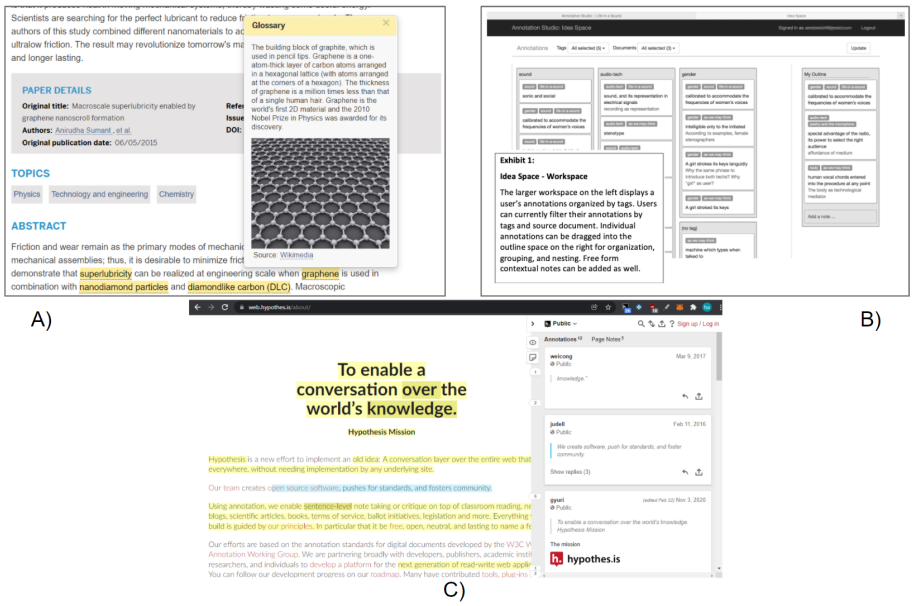


Figure 1.1: Variations on annotation clients and how the display annotations: (A) *ScienceInTheClassroom* uses a tooltip to show the annotation body besides the annotation target, (B) *Annotation Studio* uses a canvas to summarize all the annotations and (C) *Hypothes.is* shows annotations in a sidebar as a list.

reader. Differences in how annotations are created are what make practices to be heterogeneous.

1.3 Looking for an *interesting* Research Question

The process of collecting, rendering, and in general, managing annotations might be different in each scenario. Broadly, the act of annotation has some core characteristics, but at the same time, it can be highly heterogeneous, i.e., it depends very much on the context, aims, and stakeholders at play. Examples can range from theater-play investigations in the digital humanities [CM20], critical reading across course materials in education [Hyp19], fact-checking in journalism [RMSB19], or gene annotation in biology [CMP+14], to name a few. These differences explain the existence of a plethora of annotation tools (see Fig. 1.1).

The fact that these tools exist supports the idea that there are differences between different annotation approaches. At the same time, we can detect a common thread that leads us to classify all of them as annotation tools. It is this "feeling of family" that suggests that annotation tools should be developed as a "product platform" rather than as single-off products. That is, we can capitalize

on similarities between annotation tools by utilizing software reuse approaches. The purpose of this thesis is to provide proof of concept for this statement. We can define our initial RQ as

> RQ: *How would a platform for a family of web annotation tools look like?*

In terms of implementation, three main approaches can be found in annotation tools: desktop applications [GSA18, GCGdJGA+19], web-based annotation tools [SBID17], and browser extensions [Iva17]. The former is a common approach for non-web resources, web-based hosts allow users to annotate documents hosted in a centralized service, while browser extensions can annotate any web resource hosted anywhere, in a centralized service, in a third-party website, but also no-web resources hosted locally. Hence, browser extensions are a promising approach to web annotation, and we take this perspective. Browser extensions are software that adds functionality to a web browser, augmenting its native capabilities. We can then refine our initial RQ as

> RQ: *How would a platform for a family of browser extensions for web annotation look like?*

Domain-wise, when specifying a family of tools, its scope needs to be defined. Addressing a platform to support any kind of annotation tool is too wide, as W3C supports annotation of almost anything (e.g., text, images, videos) but also any workflow (i.e., how text content is reviewed or how annotations are created and used for further analysis) [Reh20]. In software engineering, domain analysis, or product line analysis, is the process of analyzing related software systems in a domain to find their common and variable parts [LKL02]. Software Product Lines are a successful approach for creating a "family of products", which share commonalities and manage a set of features (i.e., functionalities) to satisfy the specific needs of a particular domain. However, web annotation is too broad to consider it a single domain, as web annotation can include the annotation of any kind of multimedia (e.g., video, photos, geolocation, or even 3D models) or multiple annotation agents (e.g., from manual to automatic annotation using machine learning), and these contexts have little to none commonalities. As a result of that, in our case, we focus on the domain of web annotation tools for manual reviewing, specifically, the review of textual documents. Then, we reformulate the RQ as follows:

> RQ: *How would a platform for a family of browser extensions for reviewing using web annotations look like?*

The same question can be rephrased as a technological research problem using Wieringa's template [Wie14]:

How to design a platform to systematically reuse features (ARTIFACT)

that satisfies heterogeneity and extensibility (REQUIREMENT)

so that developers reduce the development and maintenance costs (STAKEHOLDER GOAL)

in the creation of web annotation extensions for reviewing? (CONTEXT)

1.4 Looking for an *important* Research Question

A research question should not only be original, but also important (i.e., relevant). This distinction is being highlighted by Dr. Medawar, Nobel Laureate in Physiology or Medicine when saying:

> *Any scientist of any age who wants to make important discoveries must study important problems. Dull or piffling problems yield dull or piffling answers. It is not enough that a problem should be "interesting". ... The problem must be such that it matters what the answer is—whether to science generally or to mankind.*
>
> (Sir Peter B. Medawar, 1979)

This quote encourages to include at least some reflection on the RQ's relevance. This section argues about the importance of the aforementioned RQ in terms of both the context where it is applied and the goal or problem that it solves. Next, we introduce the relevance of the context of annotation tools for review. Specifically, in the three reviewing cases that we have addressed in this thesis, (1) systematic literature review's data extraction, (2) feedback provision for students' assignments, and (3) peer review of scholarly research.

1.4.1 The context: annotation for review

To introduce each of the cases, we describe the context, a description of its relevance in research, and the role of annotations in this context to increase efficiency and effectiveness in their current practice.

Systematic literature review data extraction

Systematic Literature Reviews (SLR) and Systematic Mapping Studies (SMS) involve the collection and analysis of data from primary studies to answer research questions in a given field or area [KBB15]. Looking at its relevance, we have searched for the number of papers that include literature reviews in their title, abstract, or keywords. Fig. 1.2 shows the evolution in terms of publication of literature reviews in the last 30 years, evolving from less than 5,000 papers in 1990 to more than 100,000 in 2020. This means that systematic literature reviews have great acceptance and the number of practitioners is increasing. In

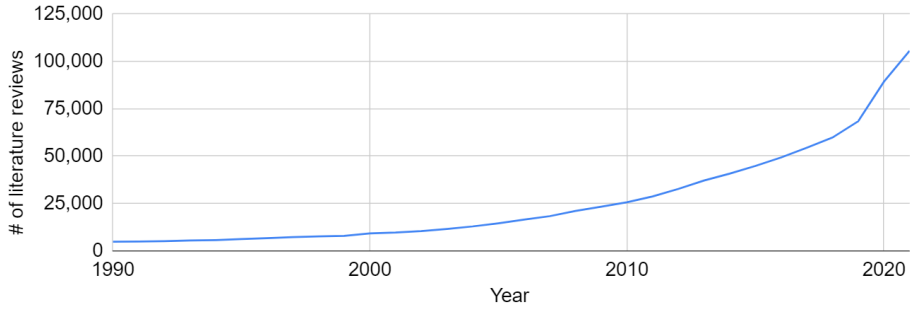


Figure 1.2: Number of publications per year where title, abstract or keywords include the term "literature review" from 1990 to 2020. Data provided by Dimensions.ai research portal.

this context, one of the most demanding stages is data extraction [GF17]. In this process, evidence is retrieved from annotating primary studies (e.g., highlighting paragraphs in the text) and assigned to a specific code within the defined classification facets).

Student assignments review to provide quality feedback at scale in Higher Education

An increased focus exists on continuous evaluation. The continuous assessment adds to grading, the concern of tracking student progress, and (re)acting accordingly. Here, feedback becomes key: the aim is not only to grade but to gain insights into students' progress. Multiple authors have noted that classroom expansion has significant implications for feedback, with increasing workloads and the requirement for quality feedback remaining constant [SM15, Wor18, Car06, HHS01]. Fig. 1.3 shows that interest in the last decades in continuous assessment and e-feedback has grown. The delivery of quality feedback implies annotating students' assignments. Thanks to web annotation, costs can be cut (i.e., via integration with Learning Management Systems or LMSs), and, consequently, reduce the costs of feedback provision to a large number of students.

Quality feedback in Peer Review

Peer review is under pressure. Demand for reviews is outstripping supply where reviewers tend to be busy people who contribute voluntarily. Authors highly value reviews, yet complain about the time it takes to get feedback to the point of putting research timeliness at stake. The interest in peer review has increased in the last 30 years, where more than 12,000 articles have been published in 2020 on this topic (see Fig. 1.4). In the context of peer-reviewing, although part of the review process has been moved to the web, the review itself is still often conducted with the only help of a yellow highlighter, physical or digital. Here,

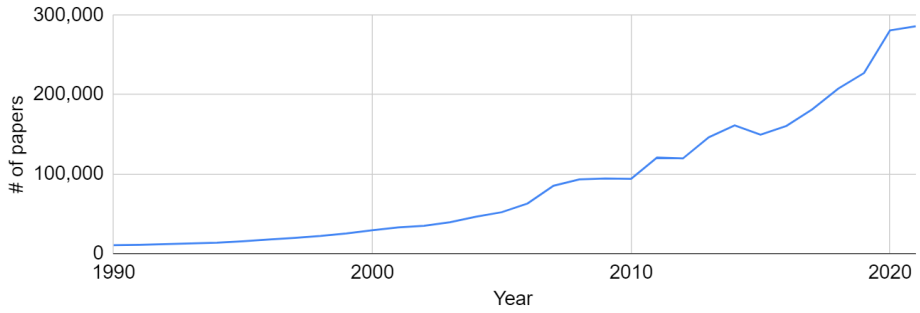


Figure 1.3: Number of publications per year about "continuous assessment or e-feedback in higher education" from 1990 to 2020. Data provided by Dimensions.ai research portal.

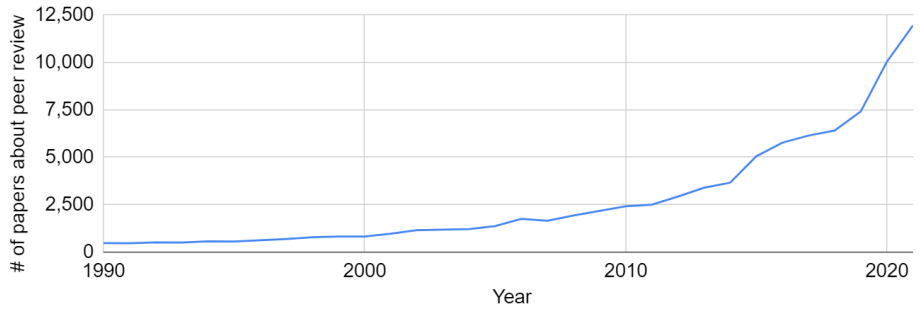


Figure 1.4: Number of publications per year including peer review in their title, abstract, or keywords from 1990 to 2020. Data provided by Dimensions.ai research portal.

web annotation tools can help increase the efficiency and effectiveness of peer review.

However, the development of annotation tools for reviewing (primary studies, students' assignments, or scholar papers) in these contexts takes a lot of effort. This moves us to the goal of this thesis: to reduce the development and maintenance costs of these tools.

1.4.2 The goal: reduce the development and maintenance cost of annotation tools for review

In 2014, the *Hypothes.is* initiative conducted a survey on the status of 60 annotation tools [1]. The results indicate that 58% of the annotation tools were available, but most of them were alpha, beta, or their use was limited. We replicated this study in 2019 to find that only 25% of the revised tools were up

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

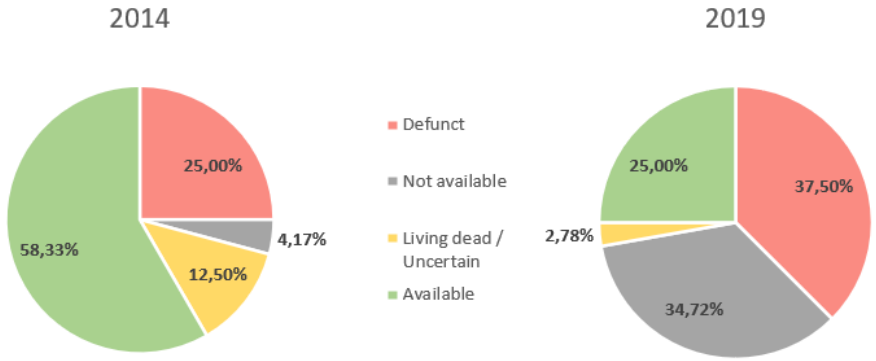


Figure 1.5: Annotation Tool Evolution: From 2014 to 2019. Available at: rebrand.ly/HypothesisSurveyUpdated2019.

and running; 35% were defunct projects and 40% were no longer available (see Fig. 1.5).

Behind these figures lies the effort involved in one-off development, which most cases end in useful but unmaintained annotation tools. Next, we analyzed a subset of 10 still-active and open-source annotation tools identified from the *Hypothes.is*' survey and previously mentioned systematic review of web annotation tools to quantify their development and maintenance cost. Results are shown in Table 1.1[2].

Development. We resort to three metrics: lines of code (*#LOC*), main contributors (*#contributors*) and commits (*#commits*). By main contributors, we refer to committers whose addend is more than 80% of the commits done in the repository. The first columns of Table 1.1 show the output. The snapshot looks as follows: annotation tool development accounts on average for 121,518 LOCs, which are contributed by around 3 developers who made almost 2,300 commits.

Maintenance. Maintenance effort is captured through the elapsed time to obtain the first stable version (*time-to-market, TTM*), and the number of commits ever since (*# commits for maintenance*). The first stable release is dated from the so-labeled v1.0 release. If there is no such label, the launch date is determined from the evidence of public release (e.g., availability at the Chrome Web Store, publication of conference papers). On these grounds, the general picture is for 18 months for the first release, and a sustained effort of almost 1,200 commits ever since.

The bottom line is that annotation tool development puts stringent demands on the tenacity and resources of the backing communities. It comes as no surprise that too many efforts deplete after a few years. Therefore, it is important

[2]Maintenance effort is captured through the elapsed time to obtain the first stable version (*time-to-market, TTM*), and the number of commits ever since (*# commits for maintenance*). The first stable release is dated from the so-labeled v1.0 release.

Table 1.1: Web annotation tool development effort. Updated in January 2021.

Tool	# LOC	# contributors	# commits	TTM (months)	# commits maint.
Hypothes.is [Hyp19]	81,540	7	8,995	60.67	3,170
Recogito2 [SBID17]	90,349	1	3,002	8.57	1,687
Annotation Press [ZNY+17]	91,694	3	990	10.30	474
WAT-SL [LKL+19]	5,824	1	20	16.53	18
Annotation Studio [Par16]	89,403	3	2,234	6.80	1,659
@note [GCSCS13]	154,726	2	262	20.20	6
Dokie.li [CGV+17]	35,724	1	3,278	49.87	1,446
Neonion [MBKB+15]	127,883	6	842	9.93	2
CATMA [Cat]	116,306	3	1,866	13.03	1,641
WebAnno [YBEG15]	197,077	2	5,199	8.93	4,362
Mean	**131,518**	**2.85**	**2,287**	**18.70**	**1,117**

to reduce the development and maintenance cost of annotation tools.

That is why all of that development and maintenance effort should be directed on lowering costs and encouraging the reuse of previous developers' work, allowing developers to focus on variability rather than reinventing the wheel.

1.5 Research approach: Action Design Research

This thesis uses Action Design Research (ADR) as its research method. The ADR research method combines Design Science Research and Action Research. Action Research aims to contribute both to the practical concerns of people in an immediate problematic situation and the goals of social science by collaboration within a mutually acceptable ethical framework [Rap70]. Design Science Research involves the analysis of the use and performance of designed artifacts to understand, explain and improve the behavior of aspects in Information Systems [IV09]. ADR is a combination of both methodologies, where the influence of the relevance cycle is stressed (see Fig. 1.6).

Sein et al. define ADR as a research method to generate prescriptive design knowledge by building and evaluating ensemble IT artifacts in an organizational setting [SHP+11]. A key insight is the role played by the organization in driving and shaping the design knowledge that ends up being instantiated in the IT artifact. Therefore, the term *ensemble artifact* denotes the artifact taking

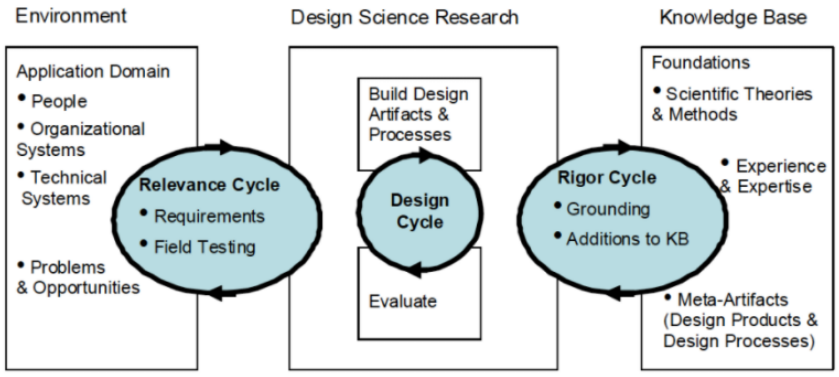


Figure 1.6: Design Science Research (DSR) Cycles (taken from [Hev07]).

its full meaning in conjunction with the context where it displays its utility, reflects the practice, and brings utility to stakeholders. Therefore, ADR conceives artifact design as a result of a researcher-practitioner collaboration within an organization. Consequently, the interaction between researchers and practitioners occurs throughout all stages. Fig. 1.7 reproduces the stages and principles of ADR [SHP+11].

Problem Formulation. The first stage is triggered by a problem encountered in practice or predicted by researchers. It catalyzes the development of a research strategy. This stage is based on two principles: *Practice-Inspired Research* and *Theory-Ingrained Artifact* [SHP+11]. The former emphasizes viewing organizational problems as opportunities for knowledge creation. Since the organization informs the solution, it is critical to describe the organization whose practices and characteristics will inform the artifact design. The second principle highlights that the intervention (e.g., the IT artifact) is to be informed by theory, the existing knowledge that grounds design decisions.

Building, Intervention, and Evaluation (BIE). This stage builds upon the problem framing and theoretical premises adopted in stage one. These premises provide a platform for generating the initial design of the IT artifact. From now on, the IT artifact is further shaped by organizational use and subsequent design cycles [SHP+11]. Or using Sein et al.'s principles: reciprocal shaping (i.e., the IT artifact and the organization feedback each other: prototypes serve to profile the interpretation of the organizational environment that help a better fit in subsequent versions), mutually influential roles (i.e., researchers and practitioners bring complementary insights), and authentic and concurrent evaluation (i.e., authenticity is a more crucial element for ADR than controlled conditions; thus assessment should take place within the company and throughout the research).

Reflection and Learning. ADR involves more than just solving a problem for an organization. To ensure that contributions to knowledge are made, conscious reflection on the problem framing, theories adopted, and the emerging

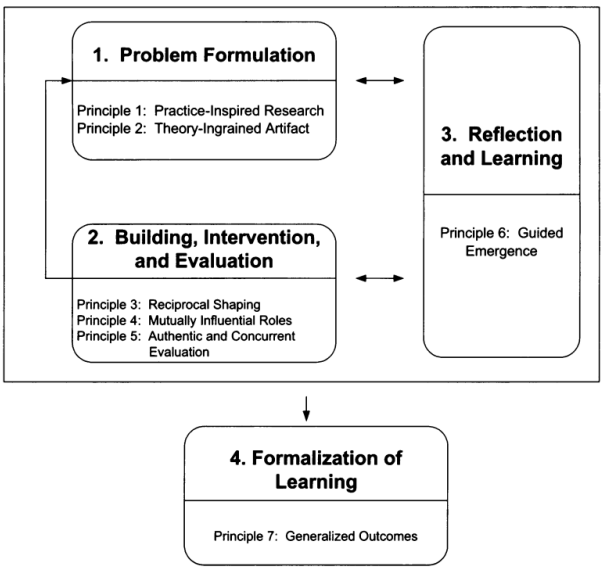


Figure 1.7: ADR Method: the first three stages form an iterative cycle which are gradually distilled into the final learnings formalized in the final stage (taken from [SHP+11]).

IT artifact are critical. The principle here is termed as *guided emergence*, where emergence captures this notion of unanticipated consequences that arise during the intervention in the organization and to which researchers should be sensitive to [SHP+11].

Formulation of Learning. At this point, we reach an ensemble artifact that brings with it some premises about the problem framing and the organization setting. It represents a solution to a problem. Both can be generalized. Sein et al. suggest three levels for this effort: (1) generalization of the problem instance, (2) generalization of the solution instance, and (3) derivation of design principles from the design research outcomes. Design principles abstract away from individual IT implementations and focus on the abstract mechanisms that provide utility and support the solution.

1.6 Outline

This section outlines the remainder of the thesis. A summary of each chapter is provided in the following lines.

Chapter 2. This chapter presents the background of this thesis. We present an introduction to web annotation and the W3C Web Annotation recommendation published in 2017. Next, we describe the variations in annotation practice and the background of the problem for the three annotation practices tackled

in this thesis. Finally, we introduce current approaches for implementing custom annotation tools and an introduction to our solution, systematic reuse of annotation features using Software Product Lines (SPLs).

Chapter 3. This chapter presents the practice of data extraction in secondary studies (i.e., SLRs and SMSs). We explore how we tackle the problem of increasing the efficiency and effectiveness of the data extraction process using a dedicated web annotation client. Taking into account the specifics required by practitioners in this domain, we have developed *Highlight&Go*.

Chapter 4. This chapter presents the second practice where web annotation is applied, quality feedback provision in Higher Education. In this chapter, we first explore the problem of providing quality feedback (specific, contextualized, personal, and timely) in large classrooms. Next, we define the meta-requirements of an annotation tool to seamlessly integrate annotation over students' documents with Moodle, a Learning Management System. Taking into account the specifics required by practitioners in this domain, we have developed an annotation client named *Mark&Go*.

Chapter 5. This chapter presents the third practice where web annotation customization is required, peer-reviewing activity in research. In this chapter, we first present the problem in peer review practice. The speed of voluntary reviewers is a major factor that affects quality review provision. Informed by the requirements to perform a good quality review, we present *Review&Go*, a customized annotation tool to support peer review.

Chapter 6. This chapter presents a process for Design Knowledge accumulation across distinct Design Science projects. This process was the one used to accumulate knowledge across the three annotation projects addressed as use cases in this thesis, where artifact development introduces reuse considerations. We advocate for the accumulation of design knowledge through the use of Product Line Engineering (PLE). PLE methodology advocates for systematic reuse by putting the focus on a family of artifacts (in this case web annotation clients) rather than on one-off artifacts. The result of this process is a Software Product Line called *WACline*.

Chapter 7. This chapter presents a description of the solution platform to face heterogeneity in annotation practices, called *WACline*. The solution is a Software Product Line that annotation tools developers can use to create browser extensions for web annotation. This chapter presents *WACline*'s platform description and its impact on feature reusability across annotation clients and the feasibility of developing new customized annotation tools at a lower cost.

Chapter 8. This chapter concludes the manuscript. To this end, key findings and contributions of this thesis are presented. In addition, limitations of the proposed solutions and future work are suggested.

1.7 Conclusion

This chapter provides an overview of the contents of this dissertation. We provided the background of the main topic, web annotation. For such a topic, we have introduced the main problem and three scenarios where annotation plays a role to increase practitioners' productivity and outcome quality.

The next chapters introduce the three practices where customized web annotation tools have been used, and the process followed to create a platform to facilitate customization of annotation tools.

Chapter 2

Web Annotation: Theme & Variations

2.1 Overview

This chapter is the background of this thesis. First, we explore the evolution from annotation to web annotation and introduce the W3C Web Annotation recommendation published in 2017. Second, we describe the variations in annotation practice and the background of the problems for the three annotation practices tackled in this thesis introduced in Chapter 1. Finally, we provide an overview of current development approaches for custom annotation tools and an introduction to our solution: systematic reuse of annotation features using SPLs.

2.2 The Theme: Web Annotation

Biblical manuscripts have liturgical notes at the margin, known as *marginalia*. Annotations are a type of *marginalia*, where notes in the margin are associated with a particular point in the document. Annotations can be anywhere: on the margin, on the side, as a footnote, between the text (inline), hanging outside the page (e.g., in a post-it), on a separate page, or at the end of a chapter (e.g., endnotes). Typically, they are placed as close as possible to the piece of information that they refer to, which can be a character, a word, sentence, phrase, a paragraph, a figure, or any other referable piece of information.

Annotation is a common (individual or social) action that conveys information, gives comments, inspires discussion, expresses power, and improves learning. It aids in the mediation of the reading-writing interaction. While reading, people underline crucial passages, identify plot devices, and remark on structure and dialogue. When writing, people refer to those passages or remark points to later write a report or a final paper. Annotation has been proved to

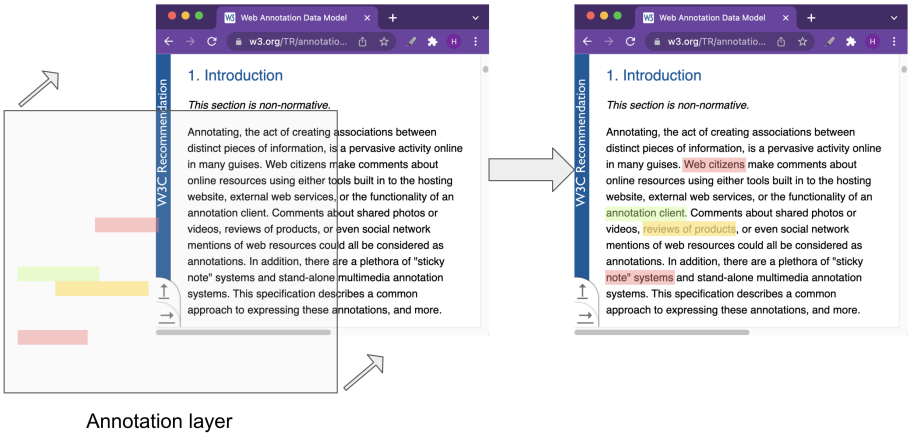


Figure 2.1: Illustration of an annotation layer over W3.org website.

be an effective way of aiding in the comprehension and interpretation of written information [KPA11].

With the advent of digitization and the web, documents were moved to the web making them available online. Web annotations (a.k.a., annotation of web resources) allow users to add, modify, or remove information from a web resource without modifying the resource itself. From the very first writings of Tim Berners-Lee, web annotations were envisioned as a layer on top of the web where end-users might complement web content with their notes (see Fig. 2.1). The very first attempt to bring annotations to the web was *NCSA Mosaic* [SH94] in 1994. Hundreds of web annotation tools have been developed since then, for general use or in specialized fields such as biology or social sciences. Due to the increase in the number of annotation systems, three attempts have been made to standardize web annotations [CSRC13]: Annotea [KK01], Open Annotation [BBC+13], and Annotation Ontology [COG+11]. Finally, in 2017, the W3C released the recommendations for web annotations [W3C17] based on these previous attempts.

W3C Annotation recommendation

The W3C Annotation recommendation is thought of as a general recommendation to describe every single type of annotation. W3C covers the whole spectrum of annotations, from annotating a fragment of an image, a text fragment, or any kind of web resource identifiable using an IRI (Internationalized Resource Identifier). The W3C Web Annotation recommendations are published in the form of three documents: data model, vocabulary, and protocol [W3C17].

The **Web Annotation Data Model** describes how web annotations are structured in a model (i.e., how data is represented) and a format (i.e., how it must be serialized) to enable annotations to be shared between systems. Fig.

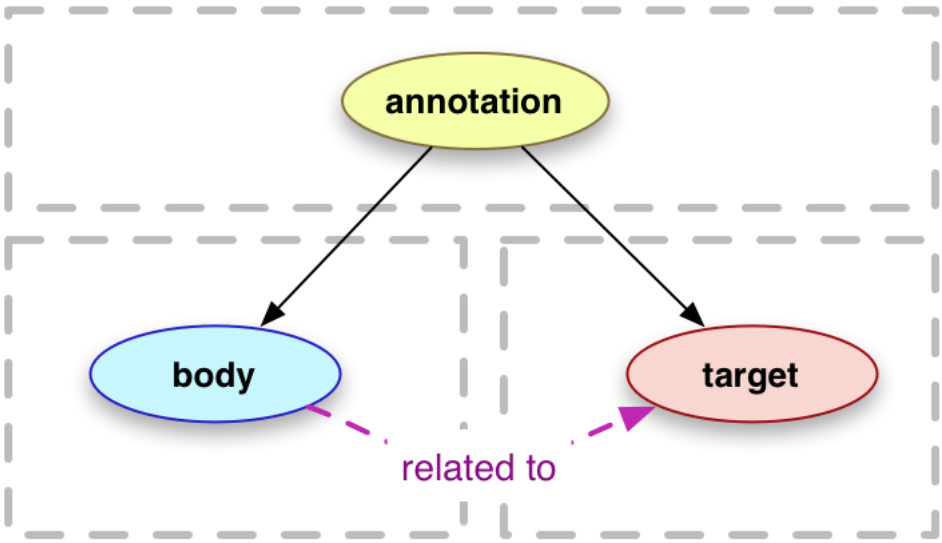


Figure 2.2: W3C Web Annotation specification.

2.2 presents the most simple annotation description. W3C Web Annotations, based on the W3C recommendation definition, are an addressable web resource (i.e., they have an IRI) composed of two main elements, body and target, and convey that the body is related to the target. The body describes what is added to the annotated resource (e.g., a comment), while the target describes which web resource (or a fragment of a web resource) is annotated. Web annotations can be serialized in JSON-LD, Turtle, or RDFa.

The **Web Annotation Vocabulary** specifies the set of RDF classes, predicates, and entities that are used by the Web Annotation Data Model. This covers the specification of properties and classes that a web annotation can have, such as creator, purpose, or selector, to name a few.

The **Web Annotation Protocol** describes the transport mechanisms for web annotations (i.e., CRUD operations) and their communication with the web architecture (i.e., REST practices).

Additionally, W3C provides two supplementary notes. The **Selectors and States** note describes which portion of web resources is annotated and the **Embedding Web Annotations in HTML** note describes and illustrates potential approaches for including annotations within HTML documents.

W3C standardizes the annotation-as-a-noun, however, it leaves the annotation-as-a-verb unconstrained, i.e., the annotation process in the pursuit of user goals. These goals can be diverse. At this point is where W3C introduces web annotation motivations and purposes (see Table 2.1).

The Annotation Model differentiates between motivation and purpose. Motivation describes why the user has created the annotation, while the purpose describes why the user has included the body (or bodies). For example, a

Table 2.1: List of annotation motivations and purposes defined by the W3C. The relationship between an Annotation and a motivation can be 0 or more, while purposes are associated with 0 or more Annotation Bodies (allowing multipurpose annotations).

Motivation	Description
assessing	The motivation for when the user intends to assess the target resource in some way, rather than simply make a comment about it. For example to write a review or assessment of a book, assess the quality of a dataset, or provide an assessment of a student's work.
bookmarking	The motivation for when the user intends to create a bookmark to the Target or part thereof. For example an Annotation that bookmarks the point in a text where the reader finished reading.
classifying	The motivation for when the user intends to classify the Target as something. For example to classify an image as a portrait.
commenting	The motivation for when the user intends to comment about the Target. For example to provide a commentary about a particular PDF document.
describing	The motivation for when the user intends to describe the Target, as opposed to (for example) a comment about it. For example describing the above PDF's contents, rather than commenting on their accuracy.
editing	The motivation for when the user intends to request a change or edit to the Target resource. For example an Annotation that requests a typo to be corrected.
highlighting	The motivation for when the user intends to highlight the Target resource or segment of it. For example to draw attention to the selected text that the annotator disagrees with.
identifying	The motivation for when the user intends to assign an identity to the Target. For example to associate the IRI that identifies a city with a mention of the city in a web page.
linking	The motivation for when the user intends to link to a resource related to the Target.
moderating	The motivation for when the user intends to assign some value or quality to the Target. For example annotating an Annotation to moderate it up in a trust network or threaded discussion.
questioning	The motivation for when the user intends to ask a question about the Target. For example to ask for assistance with a particular section of text, or question its veracity.
replying	The motivation for when the user intends to reply to a previous statement, either an Annotation or another resource. For example providing the assistance requested in the above.
tagging	The motivation for when the user intends to associate a tag with the Target.

web annotation can be created to bookmark a fragment of text on a website (*oa:motivation oa:bookmark*) and a short phrase to the body that describes what the annotated text talks about (*oa:purpose oa:describing*) can be added. W3C provides a list of motivations and purposes that can be extended: commenting, classifying, replying, etc. The differences in how annotations are created usually depend on the motivation and purpose for annotations. Commenting requires that the annotation tool provides a text box to include a comment while classifying would require support to specify and use a taxonomy [AN07]. Here is where web annotation clients' heterogeneity emerges [NŠ19].

2.3 The variations in the reviewing process

The process of collecting, rendering, and in general, managing annotations might be different (see Fig. 2.3). For example, an investigation in the digital humanities requires mostly highlighting, but not commenting or grading [CM20], while fact-checking in journalism requires grading and commenting [RMSB19]; but in biology, stakeholders require to assess correct and incorrect annotated genes, but they do not need to make textual comments [CMP+14].

In this thesis, we have addressed three reviewing practices using annotations: data extraction in systematic literature reviews, students' assignment review to provide quality feedback, and peer review of scholarly papers. Next, we highlight the goals addressed in each of the annotation practices and their variations in practice and tooling.

2.3.1 Systematic literature review data extraction

SLRs and SMSs imply collecting and analyzing data from primary studies to answer research questions in a given field or area [KBB15]. One of the most demanding stages is data extraction. The objective of this stage is to extract data from primary studies to later address the literature review's research questions. Traditionally, data extraction is realized using a data recording form (a.k.a. classification facets or codebook). In this process, evidence is retrieved from primary studies (e.g., highlighting paragraphs in the text) and assigned to a specific code within the defined classification facets). It is not rare for coding (highlighting) to be still developed on paper or, if digitally conducted, using Acrobat Reader or more sophisticated tools like NVivo. In both cases, the portability of the coded data is not apparent. For tools such as NVivo, the use of proprietary formats locks researchers into a particular tool [Eve18].

This might partially explain why spreadsheets are still the predominant tool for literature reviews [TCNK16]. Spreadsheets are easy to share not only among authors but most importantly, among SLRs or SMSs reviewers and readers. Indeed, Beck et al. highlight that spreadsheets are what is being recorded and will last as one of the main contributions of the literature review effort [BKW16]. However, spreadsheets contain the codes, but not the quotes (i.e., annotations) that sustain those codes. Open coding might be obtained if highlights become

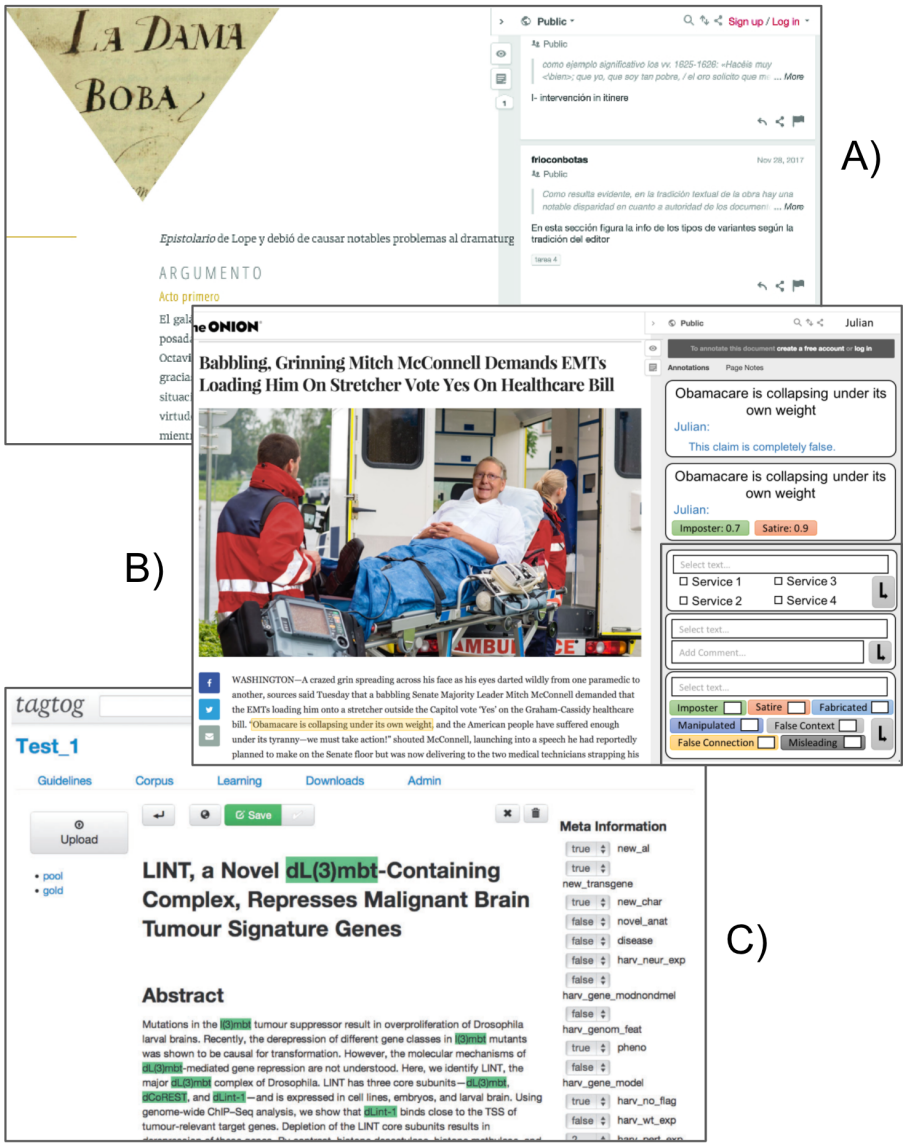


Figure 2.3: Variations on annotation clients and how the display annotations: (A) highlighting over literature, (B) commenting and grading (e.g., level of satire or manipulation) over a piece of news (C) assessing annotation of genes (e.g., using true and false) over biology papers.

web resources, and hence easily shareable. Shareable internally (as a collaborative effort among coauthors), but also externally for further validation and reuse by reviewers and third parties [PVK15].

In this context, our premise is that a customized web annotation tool may help in the practice of SLR and SMS data extraction. This leads to the following research problem:

> How to design a dedicated annotation tool
>
> that satisfies portability
>
> so that researchers conduct data extraction effectively and efficiently
>
> in secondary studies' data extraction process?

2.3.2 Providing Quality feedback at scale in Higher Education

In higher education, an increased focus exists on continuous evaluation. The continuous assessment adds to grading, the concern of tracking student progress, and (re)acting accordingly. Here, feedback becomes key: the aim is not only to grade but to gain insights into students' progress. Quality feedback is then a cornerstone of continuous evaluation whose value comes from being *timely* (i.e., provided in time to improve the next assignment), *personal* (i.e., referring to what is already known about the student), *contextualized* (i.e., framed w.r.t the assessment criteria) and specific (i.e., pointing to student's assignment) [Nic10]. The problem is not only to provide quality feedback but instead **how to deliver quality feedback at scale**. The labor of providing quality feedback for numerous students within a stringent time frame is challenging. In this context, the "Iron Triangle" between the number of students, cost, and quality [RFK19] arises. In this case, a mechanism to deliver high-quality feedback (e.g., annotations), cut costs (i.e., via integration with Learning Management Systems or LMSs), and provide feedback to a large number of students is required. This raises the following research problem:

> How to design a dedicated annotation tool
>
> that satisfies seamless integration with LMSs
>
> so that lecturers can increase the feedback quality
>
> in higher education at scale?

2.3.3 Quality feedback in Peer Review

Peer review is conducted by busy people who contribute voluntarily, where the review demand outstrips the reviewers' supply. Authors highly value reviews but complain about the time required to receive feedback from a journal or conference. Although part of the review process has been moved to the web, the review itself is still often conducted with the only help of a yellow highlighter, physical

or digital. The *Hypothes.is* initiative is promoting web annotation as a means for making peer review a truly collaborative experience [EpA17]. The stress is on the collaborative dimension of annotation, but reviewing is not *just* annotation. Peer review is governed by a reference frame that informs about what a good manuscript should contain. This reference frame underpins highlighting and commenting: reviewers look for hints within the manuscript that sustain or contradict this frame (e.g., is the significance of the problem being established?). However, this frame is domain-specific, i.e., each research methodology has its own (sometimes, tacit) checklist. To allow reviewers to provide more efficient and higher quality feedback, annotation tools should support review specifics. This leads to the following research problem:

> How to design a dedicated annotation tool
>
> that provides guidance
>
> so that reviewers can increase the feedback quality
>
> in scholarly peer review?

2.4 The variations in the implementation support

Regarding the practice of annotation, one annotation tool does not fit all. Architecturally, three main approaches can be found in annotation tools: desktop applications [GSA18, GCGdJGA+19], web-based annotation tools [SBID17] and browser extensions [Iva17]. The former is a common approach for no-web resources; annotation is made in local files (e.g., Adobe PDF Reader), while the latter web-based annotation tools and browser extensions are capable of annotating shareable documents on the web. Web-based hosts allow users to annotate documents hosted in a centralized service, while browser extensions can annotate any web resource hosted anywhere, in a centralized service, on a third-party website, but also no-web resources hosted locally. Browser extensions are a promising approach to web annotation. They work as an addition to the browser, designed to provide supplementary and customized functionality to any (web) resource that can be opened by the browser [DA15].

Browser extension approach is the case of the most well-known web annotation tools such as *Diigo*[1], *Hypothes.is* [Hyp19] or *Kami*[2]. These annotation projects are used by millions of users around the world to conduct annotation practices. Just to mention, *Hypothes.is* has recently reached 1 million users and more than 25 million annotations in education, research, and journalism. However, those annotation tools are not designed to conduct a specific annotation practice, and they provide little to no customization. The heterogeneity between different fields makes the features of annotation clients to be perceived

[1] https://www.diigo.com/
[2] https://www.kamiapp.com/

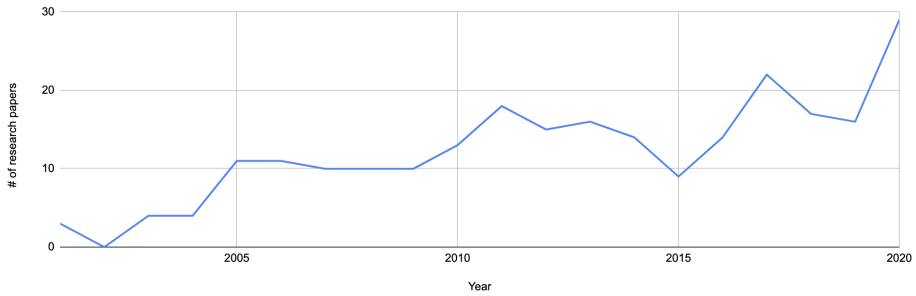


Figure 2.4: Number of publications where title, abstract or keywords include the term "Web Annotation tool" from 2000 to 2020. Data provided by Dimensions.ai research portal.

as more relevant in some areas. For example, in Social Sciences annotation tagging is not valued by practitioners for data curation [CM20] while in Biomedical Sciences it is a must [ZNY+17, RBB+15].

This situation leads practitioners to develop their annotation tools to support specific requirements in their contexts. As the adoption of web annotation increases, the number of investigations in which custom annotation tools are necessary also increases. In the last five years, up to seven different surveys and systematic reviews have been published [NŠ19, GSA18, KMK16, GCSCS18, BMB17, KTV18, CKK21] in this area. These secondary studies analyzed more than 200 annotation tools used in linguistics, education, biology, and e-health research. Web annotations popularity trend is also evident when looking at the number of researches published about this topic (see Fig. 2.4), especially since the publication of W3C Annotation recommendations in 2017.

However, as presented in Section 1.4.2, the cost of developing web annotation tools is high. To solve this, we propose moving from current implementation approaches to systematic reuse of commonalities while harnessing variabilities in the domain of annotation tools for review.

2.5 The journey towards Software Product Lines

Annotation practices share commonalities but need to be adjusted when adopted in a given research area. In most cases, developers create annotation tools from scratch. As a result, developers and researchers exert effort into developing their annotation tools to conduct their annotation practices or investigations where annotations are required. Cohen et al. [CDFF+16] reported that “In the past five years in the biomedical text mining community, we are aware of 5 projects that have developed their tools, at an estimated cost of $500,000 US" and “A significant amount of funding for research is spent on developing annotation tools in Natural Language Processing".

A frequent way to address heterogeneity is by using *Clone&Own*, where

a new software starts by cloning an existing one and then adapts parts of it to meet new requirements. *EJournalPress* [EpA17] and *FakeNewsAnnotation-Tool* [RMSB19] are examples following the *Clone&Own* approach based on *Hypothes.is*. However, although *Clone&Own* saves costs in the short run, it is not scalable if there is no way to track changes across clones.

In the same direction, current approaches favor either a configuration-based approach (e.g., *HUMAN* [WRD+20]) or the use of plug-ins (e.g., *GATE* [Ham14]). Unfortunately, current approaches to supporting all annotation practices in research are not optimal. On the one hand, a configurable product such as HUMAN includes all functionality in a single bundle, putting the stress on the user who needs to handle the diversity of configuration options. On the other hand, a plug-in-based approach (e.g., GATE) rests on an extensible architecture that allows for third parties' conjunction of extensions where incompatibilities among them might arise.

This is where SPLs come into action. An SPL approach facilitates systematic reuse of assets, in contrast to opportunistic reuse as in *Clone&Own* [Cle01]. It aims to identify common functionalities and variabilities among applications within a domain (e.g., web annotation for review), and build reusable assets to benefit future development efforts [ABKS13]. Although the costs of design and implementation at the very beginning in SPLs would require additional effort, code reuse in a *Clone&Own* setting in the long term requires additional effort. It needs to filter out code that is not going to be reused to develop the new tool, which may be time-consuming due to functionality 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 [EPPC21].

SPLs tackle the development of a whole family of software products, in our case web annotation tools for review. A major goal is to derive a product automatically from core assets based on a user's feature selection. This is realized through the sharp distinction between two interrelated processes: Domain Engineering and Application Engineering.

The main premise behind **Domain Engineering** is that most developed software systems are not new systems but rather variants of other systems within the same field. Reuse is controlled through two main activities (see Fig.2.5):

- Domain Analysis, whereby "the process by which information used in developing software systems within the domain is identified, captured, and organized to make it reusable (to create assets) when building new products" [ATFM01]. The main output is the Feature Model. Features capture "the externally visible characteristics of a system, most often by abstracting a set of functional requirements" [LKL02]. It delimits the scope of the SPL, i.e., the potential heterogeneity of the domain at hand. Accordingly, a feature model describes the characteristics of a class of products, and *not* the configuration options for a given product.

- Domain Implementation, where features are realized through reusable artifacts (a.k.a. core assets). Implementation-wise, it is common for features

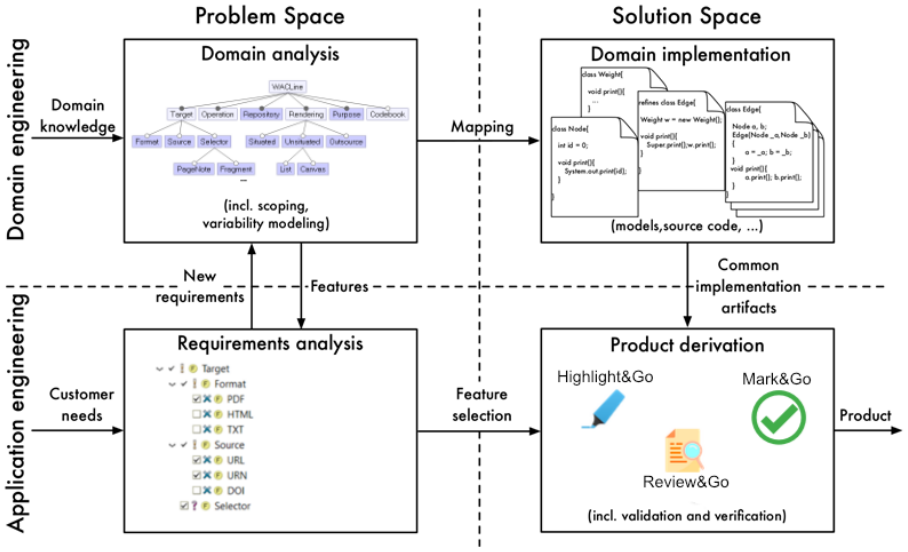


Figure 2.5: SPL development as the interplay between Domain Engineering (development for reuse) and Application Engineering (development with reuse) (adapted from [ABKS13])

to be cross-cuts; i.e., their realization is not isolated in a class or method but scattered along with different artifacts. Since features might be optional, a common approach to handle variability is using so-called *if-def* blocks.

The bottom line is that Domain Engineering does not result in a specific software product, but prepares artifacts (a.k.a core assets) to be used in multiple, if not all, products of a product line. That is why this activity is said to target *development for reuse* [Cle01]. On the contrary, **Application Engineering** targets *development with reuse*. Two steps are involved (see Fig.2.5):

- Product Configuration. In its simplest case, this step merely implies a mapping of customer's requirements to features identified during Domain Engineering.

- Product Derivation. Based on the Product Configuration, this step yields the product code along with feature-to-code mapping by selecting or eliminating named features.

SPLs are recognized as a successful approach to software variability [KPSY07]. Benefits include: reduced time-to-market [Dag00, HKM06], reduced cost [PBvdL05], and improved quality [SH04, HKM06, PBvdL05].

In annotation tools for reviewing, there is a big part of functionalities that are always shared among different annotation tools, like highlighting or storing

annotations. However, as we have seen annotation tools for review suffer from variability: annotation target (e.g., PDF, HTML web pages), annotated content consumption is conducted in different ways (e.g., creating a report) or are exported to different platforms and tools (e.g., to a wiki). Therefore, SPLs might be a good treatment to bring the benefits of systematic reuse to annotation tool development.

2.6 Conclusion

This chapter provides an overview of what annotations are, how they have evolved into web annotations with the advent of the web and how they should be described and transported based on W3C recommendations. However, W3C does not restrict the annotation-as-a-verb, that is, the annotation process in the pursuit of reviewing goals. These goals can be diverse in terms of annotation purposes, which require customization of annotation tools. We describe commonalities and variations in annotation among the three review practices addressed in this thesis. Finally, to reduce the development and maintenance cost of those tools we propose the use of SPLs and describe what this software engineering method is.

Chapter 3

Web annotation for Data Extraction in Systematic Literature Reviews: *Highlight&Go*

3.1 Introduction

Systematic Literature Reviews (SLR) and Systematic Mapping Studies (SMS), also known as secondary studies, are literature reviews that have been extensively used in software engineering to identify clusters of related studies and research gaps [KBB15]. According to a recent review [AZCHH17], one of the most challenging steps during literature review elaboration is Data Extraction (DE), i.e., extracting the required data from the Primary Studies (PS). DE is challenging in terms of human effort and time consumption (i.e., efficiency), quality of data (i.e., efficacy), and data management (i.e., data portability). However, only 20% of the literature review tools support the DE task to some extent[1].

Usually, data extractors resort to spreadsheets to manage extracted data [TCNK16]. Since spreadsheets are not specialized literature review tools, extraction is performed manually by annotating primary studies on paper or digitally, but extractors only register data perceived as essential. For instance, data extractors tend to register the classification code that typifies the primary study, but they do not gather "quotes in the text" or pieces of evidence that sustain the classification decision as it requires a lot of effort.

Qualitative Data Analysis Software (QDAS) packages for literature reviews collect some required data (including "passages in the text") but usually in

[1] 50 tools out of 242 support DE according to http://systematicreviewtools.com/ 2021 December

proprietary formats, hence hindering data portability. As O'Connor et al. state "interoperability remains an urgent need, and with each new tool that is not compatible with other systems, that need becomes more pressing" [OTG+19]. This limitation causes researchers to be locked into a particular tool or increases the time required to move data from one tool to another due to a lack of data portability [AZCHH17].

In this context, we look into web annotation tools to support efficient, effective, and portable data extraction for secondary studies. Hence, the main premise of this chapter is that

> *W3C's Web Annotations can be used to support data portability, creating a seamless integration between primary studies and spreadsheets to make data extraction more efficient and effective.*

This chapter introduces the practice and problems of data extraction in secondary studies, introducing the specific case of the Onekin Research group of the University of the Basque Country. Then, following the ADR method, we introduce an annotation tool for color-coded highlighting of primary studies that integrates with Google Sheets. The aim is to improve the efficiency of conducting an effective data extraction process for SLR and SMS authors, but also to benefit third-party stakeholders (e.g., journal reviewers and readers) by increasing its portability.

3.2 Problem formulation

ADR reflects the premise that IT artifacts are ensembles shaped by the organizational context during development and use [SHP+11]. Therefore, it is most important to analyze the characteristics of the actual practice that will inform the design of the artifact. This section aims to contextualize the main problems faced in data extraction departing from our own research experience.

3.2.1 Practice-inspired research

Our research group, Onekin, is prolific in research, reporting four published systematic reviews [PD19, AMD21, MD16, DA15] in recent years. Table 3.1 abridges figures from systematic reviews. Regarding secondary studies, researchers in our group have found several difficulties. The motivation to face such difficulties is to boost secondary study production in our research group. We based our argumentation on the experience of performing those four secondary studies. Secondary studies are undeniably time and resource-demanding. From our experience, the main conclusion is that data extraction is a critical and difficult step that is susceptible to improvement. The authors of the literature reviews in our group reported concerns about dealing with multiple services/tools and performing non-added-value tasks. In a normal case, a data extractor has to access a journal website, download the PDF of the primary study, store the file in a Dropbox folder or a Mendeley account, open the PDF

Table 3.1: Description of Onekin's Systematic Reviews.

Systematic Review	[DA15]	[MD16]	[PD19]	[AMD21]
Type	SLR	SMS	SLR	SMS
Guidelines	[KC07]	[PFMM08], [PVK15], [KC07]	[PFMM08]	[KBB15]
# researchers	2	2	2	3
# primary studies	42	107	30	66
Extraction strategy	Independent	Ph.D. student + supervisor	Independent	Independent

using Adobe Reader or through an application or browser-based reader, read the paper underlining the primary study's quotes and, finally, copy&paste these quotes to a spreadsheet and classify the primary study into the spreadsheet. This illustrates the loss of time in non-productive tasks (i.e., downloading PDFs and copy&pasting quotes) and the low support of the technology, specifically, the lack of automation and/or tool interoperability. In a first attempt to solve some of these problems, one of our researchers maintained a mind map with a list of links to papers so that she was able to browse in an agile manner when he wanted to read them. One step forward was a script developed by another researcher to automatically scrape journals' websites, generating an Excel of primary studies' metadata including links to download PDFs. However, the bulk of the work of reading primary studies and gathering evidence remained unsolved. Hence, we drove our research to a deep analysis of the data extraction task with the challenge of reducing the waste of time.

Independent data extraction activity

Kitchenham et al. [KBB15] described guidelines for conducting evidence-based systematic reviews in software engineering. These guidelines are divided into three main phases: plan review, conduct review, and document review. First, the planning phase specifies the research questions (RQ) to be addressed in the review and defines a review protocol (i.e., a classification scheme) to extract data to answer the RQs. Second, the conducting phase performs the work by searching and screening primary studies, distilling the classification scheme, extracting data from primary studies, checking data extraction, and drawing conclusions through the synthesis and analysis of the extracted data (see Fig. 3.1). Finally, the reporting phase describes the literature review and its conclusions in an academic paper. This work focuses on independent data extraction activity. Although multiple data extractors may participate in the data extraction, guidelines recommend that each extractor works independently to avoid

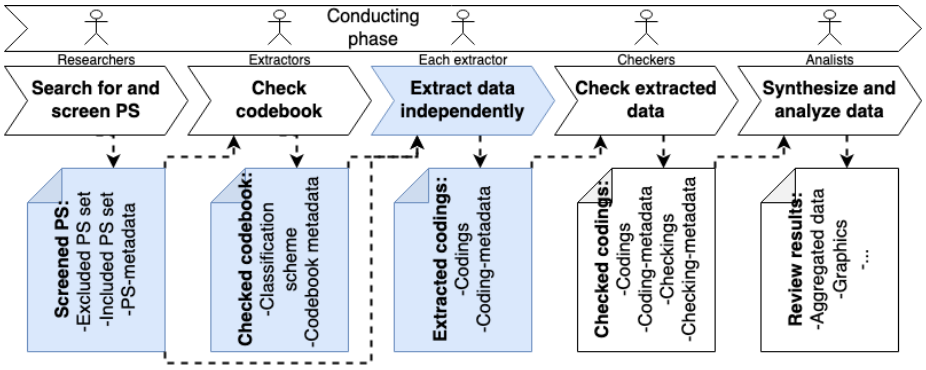


Figure 3.1: Conducting phase diagram showing activities, roles and deliverables. Data extraction and its input and output deliverables are shown in blue shading. Note that phases are not strictly linearly followed but are iterative steps.

undesirable biases.

Tools for data extraction

Tools for data extraction should help extractors in their work. However, according to the Systematic Review Toolbox website[2] (SRT) only 20% of the registered literature review tools (10% in the Software Engineering realm) support data extraction. In addition, few tools offer functionality for every step of the literature review. Only Buhos [BMSD18] supports all but one (automated analysis) feature listed by SRT. Therefore, **there is not a "one size fits all" literature review tool**, forcing researchers to use different tools and port data (mainly manually) among them when progressing through the literature review steps. Indeed, the International Collaboration for the Automation of Systematic Reviews (ICASR) identifies the **need for conceptual and infrastructural compatibility** to integrate different literature review tools into systematic review workflows, which would be reached through the development of an open API [OTG+19].

Problems of the current practice

Summing up, SLR tools and the data extraction activity establish a dismal background. On the one hand, there is not a universal SLR tool suitable for data extraction, which forces data extractors to move data among tools. However, the tools provide limited import and export functionality, even though data extraction activity relies on data from previous SLR stages (i.e., the codebook and the PS to classify). On the other hand, data extraction is a long process involving multiple tasks which produce a high mental workload exacerbating

[2] http://systematicreviewtools.com/

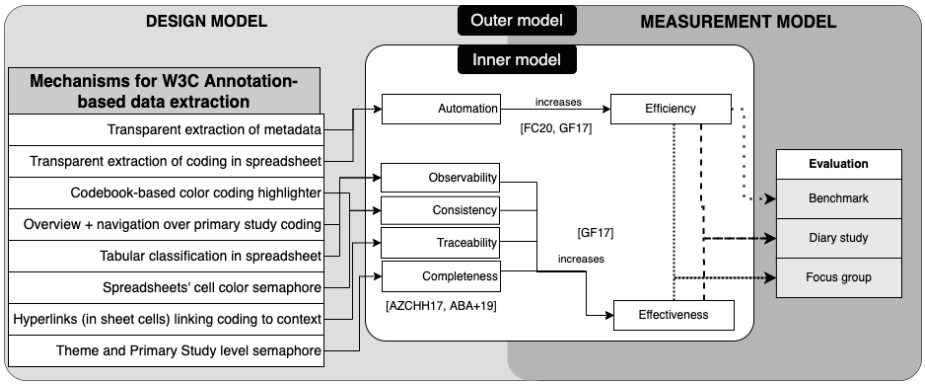


Figure 3.2: Adaptation of Inner-outer model [NO16] for the problem described in project *Highlight&Go*. Inner-model describes the independent (e.g., observability) and dependent (e.g., effectiveness) variables for the problem in data extraction activity. The outer-model describes the mechanisms implemented that influence independent variables (e.g., transparent storage increases automation) and evaluation measures the dependent variables (e.g., efficiency is measured by conducting a benchmark).

data extractors' bias, fatigue, and lack of time [CHHK13, BZ09]. These factors produce data extraction errors and affect data quality. Based on impressions gathered by practitioners in our research group, we classified the problems found into two main categories:

- Efficiency problems: literature reviews take a substantial time to complete depending on the topic, ranging from months to years. DE is *per se* a long and time-consuming activity due to multiple tasks and several PSs to be read. Limited support of literature review tools hampers time reduction since extractors have to resume and switch tasks manually, increasing the non-productive periods.

- Efficacy problems: it is critical for the literature reviews' analysis phase to provide a correct set of codings to classify PSs. For this purpose, the data checking step reviews the coding to detect and correct errors (i.e., coding inconsistency and coding incompleteness). As Garousi et al. state "quality of a literature review depends on the quality of the data and the effectiveness of the data extraction activity" [GF17]. Therefore, it is important to obtain error-free data on DE activity. However, DE is an error-prone activity caused by human fragility. Humans are in charge of dealing with data reconciliation, data integrity, and data loss due to the lack of tool support.

This leads us to the following research problem:

How to design a dedicated annotation tool

that satisfies portability

so that researchers conduct data extraction effectively and efficiently

in secondary studies' data extraction process?

3.2.2 Theory-ingrained artifact

This work is based on three main theories that are introduced in the inner-outer model (see Fig. 3.2): the use of spreadsheets as a purposeful artifact to increase efficiency and effectiveness [GF17], automatizing spreadsheet feeding [OTG+19], and the theory of standardization of extracted data to support portability [YYH+12, BCBK17].

Automation to support efficient and effective data extraction

Several studies have addressed the problem of efficiency and effectiveness in DE (see Fig. 3.2). One of the most important facts that led us to this situation is the lack of tool support for DE [HCHAZ16, AZCHH17]. Garousi et al. [GF17] present experience-based guidelines to reduce time-consuming and error-prone DE. For data recording and logging (a.k.a. coding), they have used *Google Sheets* incorporating traceability to pieces of evidence. To trace pieces of evidence, color coding annotations are used, where each of the higher-order themes has assigned a color (see Fig. 3.3). However, all this work is manually done, and automation might reduce time investment while keeping the effectiveness of Garousi's approach [FC20]. One of the aspects we found in our practice that does not provide added value and takes time is manual data logging (i.e., translating decisions from papers to spreadsheets). Even traceability facilitates spotting problems, manually conducted logging makes extractors provide incorrect categorizations (i.e., lack of consistency) or lack of completeness, and those inconsistencies can put threats to the validity of the secondary study [ABA+19]. To solve this, tool support should facilitate observability of inconsistencies and incompleteness. Spreadsheets are good to have an overview of literature reviews.

However, it should be manually defined with some formulae to easily spot those problems, and this is where automation should come into the spreadsheet view. To facilitate interoperability between the reading realm (i.e., the paper in PDF) and the recording realm (i.e., the *Google Sheet*), we resort to web annotations. The benefits of web annotation are twofold. On the one hand, web annotations can be processed to automate the recording and logging of a spreadsheet (e.g., *Google Sheets*) via its API [OTG+19]. On the other hand, web annotations are referable resources (i.e., with its IRI[3] and they can be used to trace back from Spreadsheets back to the paper.

[3]IRI: Internationalized Resource Identifier https://www.ietf.org/rfc/rfc3987.txt

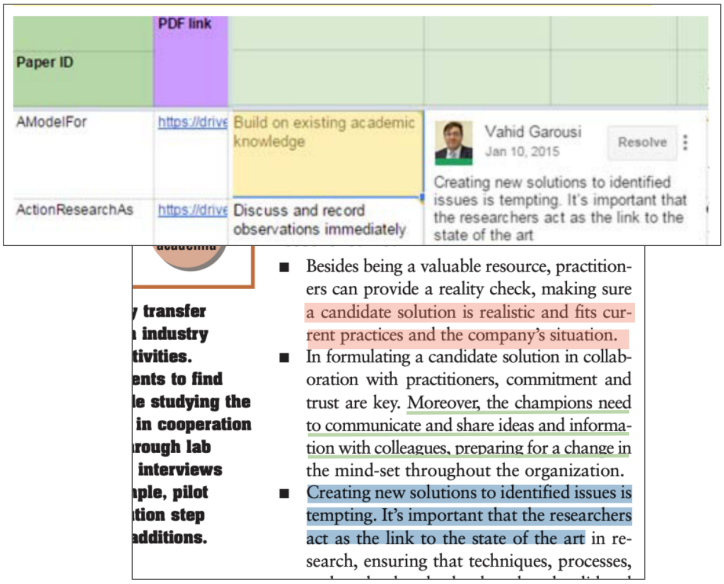


Figure 3.3: Garousi et al.'s proposal of how to extract data from primary studies to Google Sheets to conduct data extraction. Adapted from [GF17]

Standardization to support portability: W3C's web annotation recommendations

Data extraction in secondary studies is a type of data curation. 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 [CCW16]. Curating data requires annotating, publishing, and presenting data available for reuse and preservation [ADNF07, GST+02]. Literature reviews are a type of data curation where data from primary studies is organized and managed for the research community. In the same way, data in secondary studies should be reusable and preservable [ZSJ20]. This can be solved by using web annotations.

Annotation 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) [CCL06]. In the same way, web annotation has been proposed as a successful mechanism to reuse curated data over online research [KKPW21], where currently most of the primary studies reside. In this work, we propose web annotation as a mechanism to support data extraction. We followed the W3C Web Annotation recommendation as it makes data portable. Data portability means that data can be easily preserved and reused by third parties.

In literature reviews, pieces of evidence are gathered to help answer the research questions. Highlighting is an annotation made on specific paragraphs of the document. To single out those paragraphs as the target of the anno-

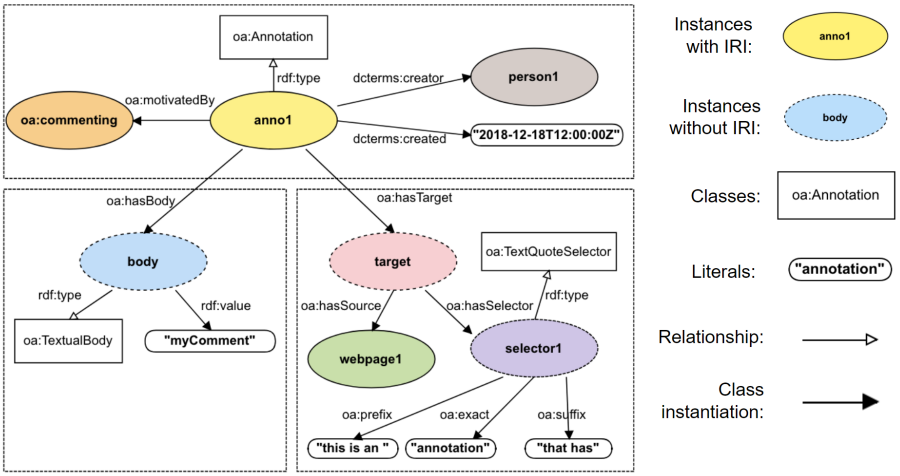


Figure 3.4: An annotation data model where the text fragment "*annotation*" in the target *webpage1* is commented with the comment "*myComment*".

tation, W3C provides a mechanism called *Selector*. Fig. 3.4 provides an example: *oa:hasBody* stands for comment "*myComment*"; *oa:hasTarget* points to an *oa:SpecificResource* resource pinpointed through the quote *"annotation"* that appears in source *webpage1*. In the example, the way to single this quote out is by indicating the text that precedes "*this is an*" and follows "*that has*" the text paragraph that is the focus of the annotation. In addition, W3C provides properties to indicate the annotation's provenance (*dcterms:creator*)[4], when the annotation is created (*dcterms:created*) or the reasons why the annotation was created (*oa:motivatedBy*). W3C includes a predefined list of motivations, which is possible to extend with new more precise motivation definitions. The next section resorts to this capability to account for literature review coding.

3.3 Building, Intervention, and Evaluation process

We have tested out the theory through a purposeful artifact developed in an ADR setting in two iterations (see Fig. 3.5), where two researchers participated in the design process of the annotation tool for data extraction while testing it in a real environment (i.e., data extraction as part of a mapping study in chatbots in e-health [PD19]). Finally, the resultant annotation tool has been evaluated by three researchers, two of them from the Department of Languages and Computer Systems of the University of the Basque Country and one from

[4]dcterms: This alias identifies the namespace of the Dublin Core Schema. This schema defines a set of vocabulary terms that can be used to describe digital or physical resources.

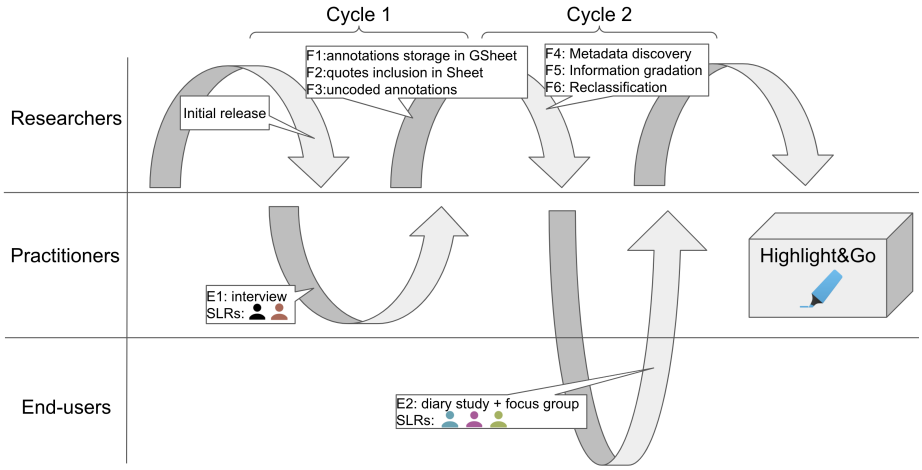


Figure 3.5: Evolution of the *Highlight&Go* project. Y axis stands for the team members (researchers and practitioners) and end users. X axis stands for the evolution in time along with the two main cycles.

the Tecnalia Research Center. The three researchers used the tool to conduct DE in their secondary studies, where a diary study evaluation process was followed and a confirmatory focus group was conducted to gather qualitative opinions about the use of *Highlight&Go* and its impact on efficiency and effectiveness. In the same way, a quantitative comparison between the purposeful artifact and Garousi-like spreadsheets has been made to measure the efficiency in terms of user interactions to conduct DE [GF17].

We start by building the solution, *Highlight&Go*, a browser extension that supports data extraction using web annotation and *Google Sheets*. Next, we describe what kind of data is needed to be recorded in the data extraction process to make it portable and how it can be done following the W3C Web Annotation recommendation. After that, we introduce how *Highlight&Go* supports this recommendation and facilitates data extraction for researchers.

The next sections delve into the details. We start by describing how the building phase was conducted, to account for data portability and data extraction efficiency and effectiveness.

3.4 Building: Acting on data extraction portability

Data extraction is "the process of examining and organizing the data contained in each study of the systematic review. It involves identifying one or more passages in the text that exemplify the same theoretical or descriptive idea" [CD11]. Broadly, the output of data extraction is a set of *coding tuples*: $<pa-$

per, category, extractor, code, quote, validation> that account for the act of a *extractor* classifying a given *paper* along a certain *code* for the *category* at hand on the grounds of some paragraphs or *quotes* found on this paper. In addition, a data extraction checker (i.e., a supervisor or a manager for the literature review) might conduct the *validation* of the mapping decisions of the extractors [BKB+07, SN07].

Coding tuples are not obtained in a single step, but they are gradually elaborated. Cruzes et al. identify the Integrated Approach as the most relevant for coding practices in the literature review [CD11]. Here, codes can be obtained bottom-up from quotes in the primary studies (inductive approach), but codes can also be readily available from previous studies where codes are grouped into categories (deductive approach). Ideally, it is recommended to reuse existing categories, as this allows for comparability between studies [PVK15]. Categories for the "research approach" introduced in [WMMR06] are a case in point, where Wieringa et al. catalog research types as "Conceptual proposal", "Evaluation research", "Experience paper", "Solution proposal", and so on. However, categories are not always available, and reviews of the literature must often introduce their classifications [PVK15]. In this case, an "open coding" is being suggested [PFMM08]. In the beginning, a set of paragraphs are identified that are coded after the research question. In this way, several quotations are obtained from a set of pilot studies. These initial quotes need to be further elaborated until a set of codes emerges that properly accounts for the distinct evidence found in the pilot studies.

The previous paragraphs identify distinct activities that intertwine during the gradual obtention of the <*paper, category, extractor, code, quote, validation*> tuples. First, **"codeBookDevelopment"** where the *codes* are introduced (terminology along reference [MML98]). Second, **"categorization"** where *category* codes are created by defining links between codes. Third, **"classifying"** where *paper* is characterized along with a *code* based on some *quotes*. To stick with the ontology of W3C Web Annotation [5], differences among those annotation efforts are captured as distinct *oa:motivatedBy* values.

3.4.1 Classifying

Fig. 3.6 illustrates the case of the mapping of the *primaryStudy1* with code *annoCodeEvaluationResearch* on the grounds of the *"we focus on empirical evaluation"* quote. The motivation of the annotation is set to *oa:classifying*, and this text segment is coded by *dcterms:creator* "*reviewer1*" . Worth noticing that the code is not a value but an annotation itself, which makes it possible to be referred to. This possibility moves us to describe how codes can be described in terms of web annotations in the codeBookDevelopment activity.

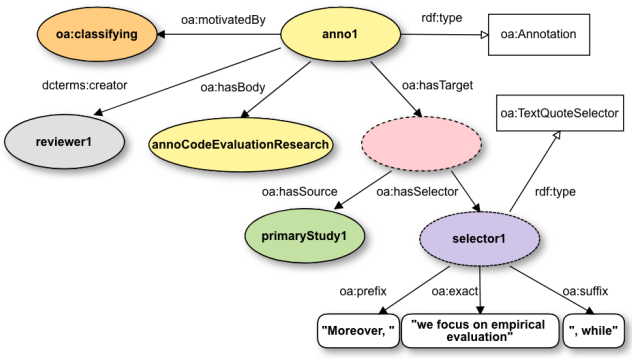


Figure 3.6: ***Classifying*** reframed as the process of annotating with motivation *oa:classifying*.

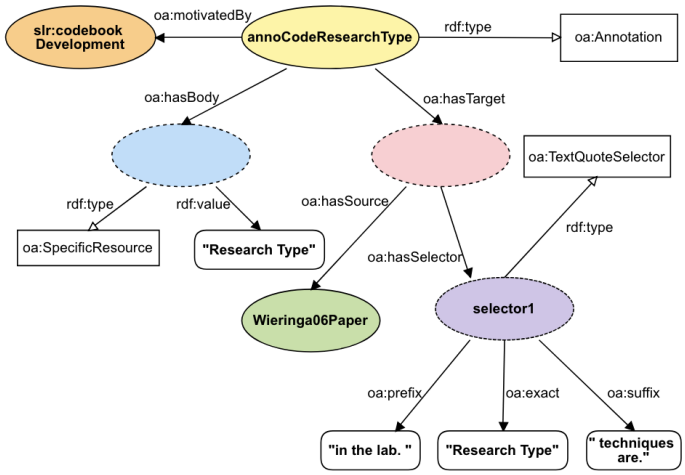


Figure 3.7: ***codeBookDevelopment*** reframed as the process of annotating with a *slr:codeBookDevelopment* motivation.

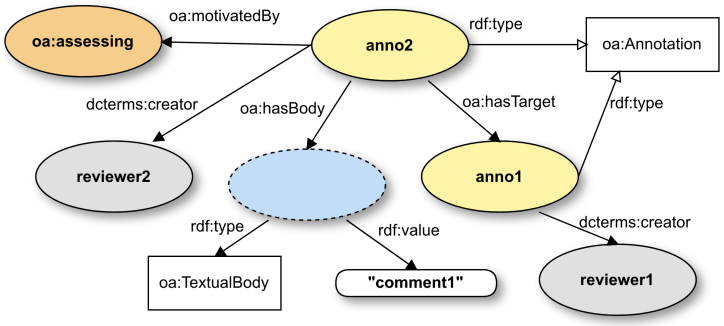


Figure 3.8: ***Assessing*** reframed as the process of annotating with motivation *oa:assessing*.

3.4.2 CodebookDevelopment

Codes are introduced also through annotation. However, now the target of the annotation is not a primary study, but a *reputed paper*. A reputed paper defines the code being introduced. Fig. 3.7 shows the case of the "*Research Type*" code that annotates *Wieringa06Paper*, specifically the textual paragraph where the word/definition appears. The motivation is set to *slr:codeBookDevelopment*. This is an extension of the W3C annotation's motivation presented in Section 2.2. This extension is described in `https://rebrand.ly/ease19ontology`.

3.4.3 Assessing

Good practices advise that initial coding of data should be revised [KBB15]. Annotation-wise revision can be considered a meta-annotation process where extractors annotate (inform) upon other annotations (classification annotations). Fig. 3.8 illustrates this situation: *anno1* stands for the annotation described in Fig. 3.6; *anno2* is a new annotation that comments "comment1" (*oa:hasBody*) on top of *anno1* (*oa:hasTarget*). This new annotation is performed by *reviewer2 (dcterms:creator)* with the purpose of validating (*oa:assessing*) *reviewer1's anno1* annotation.

3.4.4 Categorization

For our purposes, categorization is the process of upgrading a code as a higher order theme. This implies setting some structure among the code set which is captured as code relationships: *codeA* is a category for *codeB*. Therefore, *codeA* and *codeB* already exist, and categorization is operationalized as establishing a link from *codeA* (i.e., the category) to *codeB* (i.e., the enclosed code). Fig. 3.9 shows an example: *annoCodeResearchType* code is turned into a category by setting an annotation where *annoCodeResearchType* is the *oa:hasBody*, and

[5] https://www.w3.org/ns/oa

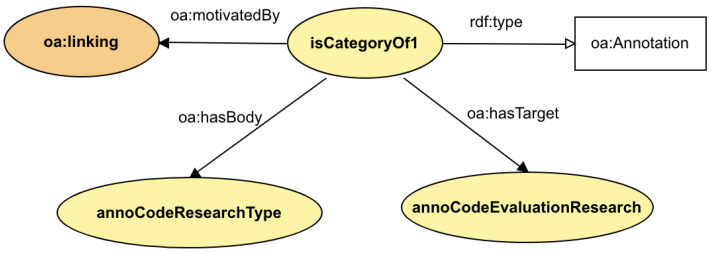


Figure 3.9: ***Categorization*** reframed as the process of annotating with an *oa:linking* motivation.

annoCodeEvaluationResearch is the *oa:hasTarget*. Both codes should already be defined during codeBookDevelopment. Here, we reuse W3C's existing motivation: *oa:linking*.

In the next section, we will present the different mechanisms implemented to reach the W3C data model by end-users making DE more efficient and effective.

3.5 Building: Acting on data extraction efficiency and effectiveness

Highlight&Go combines a browser extension and a Google Sheets script [6] that works jointly to support data extraction in secondary studies. The browser extension acts as the tool to write (capture data) while the spreadsheet works mainly as the tool to read (visualize the captured data) (see Fig. 3.2). First, we will introduce the browser extension.

3.5.1 Write: Highlighter

Highlight&Go browser extension is a chrome-based extension available in the Chrome Web Store[7].

Highlight&Go has implemented four "writing" mechanisms to support data extraction in primary studies: the codebook-based color-coded annotator, the transparent feeder to persist annotations in Google Sheets, the in-context overview and navigation interface, and the automatization to capture primary studies' metadata.

Codebook-based color-coded primary study annotation

Highlight&Go browser extension permits color-based text annotation in PDF and HTML documents on the browser. Colored themes and codes in the code-

[6] A Google Apps script injected in the spreadsheet itself which extends Google Sheets' default functionality: https://developers.google.com/apps-script/guides/sheets

[7] *Highlight&Go* is available to download at https://rebrand.ly/highlightAndGo

book facilitate the classification of primary studies in the document. To this end, the user has to select the text paragraph that supports the code decision and click on the corresponding code button on the left sidebar. Fig. 3.10 shows an example case using the paper titled "SPLEMMA: A generic framework for controlled-evolution of software product lines" as the primary study. In this example, the reviewer has classified the paper using the red color, which refers to the theme *Asset Type*, with the code *Code assets*. The number in the left sidebar denotes the number of evidence (i.e., annotations) found for that classification. Additionally, it is possible to add comments to each annotation that act as a memo. This can be useful to remember decisions or aspects a reviewer has taken into account for the classification decision apart from the highlighted evidence (i.e., quote in the document).

In the example in Fig. 3.10, mapping has been conducted for each of the four themes that characterize evolution in Software Product Lines: *Asset type* (red), *Evolution activities* (pink), *Product-derivation approach* (blue), and *Research type* (yellow). To define these four themes and their enclosed codes, it is as simple as clicking on the "create new theme" button and describing (name & description & whether it is multi-valued) each of the themes and the corresponding codes in the highlighter (see Fig. 3.11). This must be done only once at the very beginning of the data extraction activity by one of the participants in the secondary study. However, it is common in secondary studies, while reviewers curate primary studies to make slight changes in the codebook (e.g., adding a new code inside a theme). *Highlight&Go* supports these modifications at any time. Furthermore, it is possible to create a theme or code referring to reputed papers. Operationally works in the same way as classification, where the name or definition of the theme in the reputed paper is selected, and then "create new theme" is clicked. Cross-references to reputed papers or a description attached to a theme or code can be consulted by reviewers at any time directly in the sidebar. These descriptions and references reduce misunderstandings of the codebook (reducing possible individual bias and consequently increasing consistency), and they help retrieve the meaning of codes, speeding up the classification activity.

Transparent extraction of metadata

Highlight&Go provides automatic capture of primary study metadata. Kitchenham et al. [KBB15] recommend that the main author collect publication details for each of the primary studies. These data include authors' names, primary studies' titles, publication venue, publisher information, etc. *Highlight&Go* has implemented a mechanism to automatically collect in a spreadsheet some publication details (title, authors, DOI, and publisher) for each of the annotated papers. It is done transparently and automatically by the tool after the first annotation is done over a document. This metadata is later captured, thanks to the persistence mechanism (see later in this section), in a sheet of the *Highlight&Go* generated Google Sheets (see Section 3.5.2).

In addition to capturing publication details, since *Highlight&Go* annotates

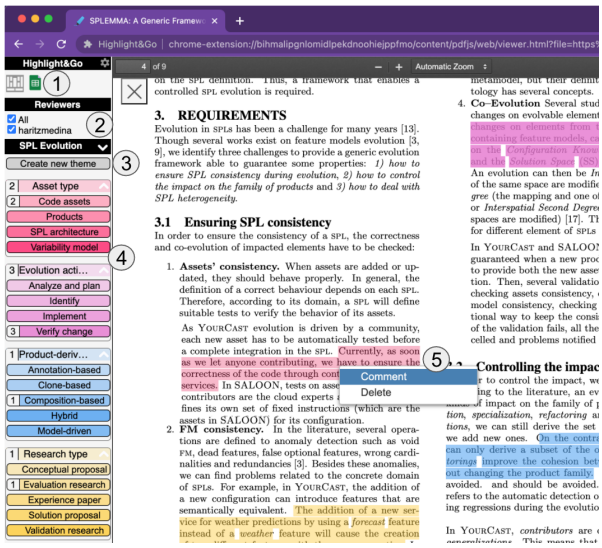


Figure 3.10: *Highlight&Go*'s highlighter user interface to conduct color-coded annotation. (1) Toolset to navigate to visualizations in spreadsheets, (2) user filter in current document, (3) codebook selection and definition, (4) color-based highlighter and (5) annotated text with the corresponding color of the selected codebook button and the possibility to add a comment or memo.

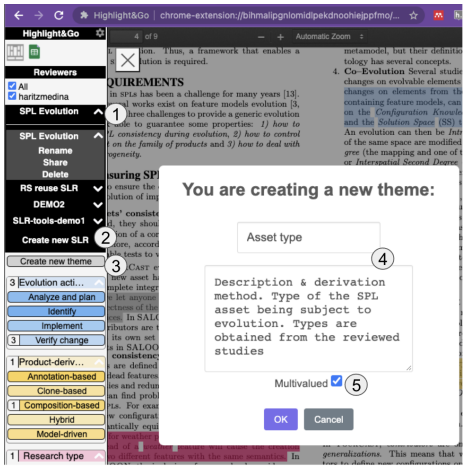


Figure 3.11: Codebook development in *Highlight&Go*. (1) Selector of current literature review, if the user is enrolled in more than one. (2) Button to create a new literature review (what creates a new spreadsheet). (3) Create a new theme button. Right-clicking on a theme makes it possible to create a code. (4) Name and description of the new theme to include in the codebook. (5) Whether the new theme is multivalued (can be classified with more than one code) or not.

web and PDF documents, the reliability of the annotation is a must. This means that annotations can be done over a PDF document locally, on the web, or over an HTML document in the publisher's digital library. Annotations' target can differ in its URL (where the document is located) or URN (the content representation of the document, e.g., HTML or PDF). To solve this, *Highlight&Go* is document agnostic, as it identifies different documents as the same one using its URN, URL, DOI, or a combination of them. The mechanism is inspired by *Hypothes.is* DOI federation [Ude17]. However, to increase its reliability, we have enhanced this *Hypothes.is*'s mechanism with the addition of three heuristics for DOI discoverability:

- Website metadata capture: some digital libraries and PDF documents include the metadata of the paper. Metadata, to be recognizable, should follow any kind of standard like Dublin Core ontology, Open Graph tags, etc. It is not that common for PDFs to include metadata about the paper, because it depends on how the PDF was created (by the publisher, which usually includes, or individual authors, which usually do not include). Digital libraries such as ScienceDirect or ACM include inside the <head> tag metadata, which is captured by *Highlight&Go* to identify the research manuscript.

- Digital libraries' website scraping: *Highlight&Go* has some heuristics to find DOIs in the most popular Digital Libraries (Springer, ACM, IEEE, and ScienceDirect). To capture the current document's DOI, *Highlight&Go* looks for the most common places where the DOI can be found in the DOM using XPATH or CSS selectors.

- DOI website browsing tracking: this mechanism is activated if the previous two fail. Some websites do not provide DOIs in their metadata. Neither is it supported by *Highlight&Go* scraping, doi.org acts as a URL resolver, redirecting the browser to where the paper can be found. In these cases, if the user navigates to the primary study using a doi.org-like URL, *Highlight&Go* keeps track of this redirection, matching the visited document (PDF or HTML website) with its DOI.

In the same way as *Highlight&Go* provides additional mechanisms to capture documents' DOIs, captured metadata is also more reliable, as the DOI is used to retrieve the rest of the publication details (authors, title, publisher, etc.) from the DOI API.

Overview&detail interface on the primary study realm

To facilitate cross-checking, *Highlight&Go* supports in-context validation with two functionalities: highlights filtering and replying, and voting functionality. After more than one reviewer extracts data independently, *Highlight&Go* joins annotations made by all extractors. The sidebar includes an extractor filter to hide/show annotations. This facilitates navigation through the rationales

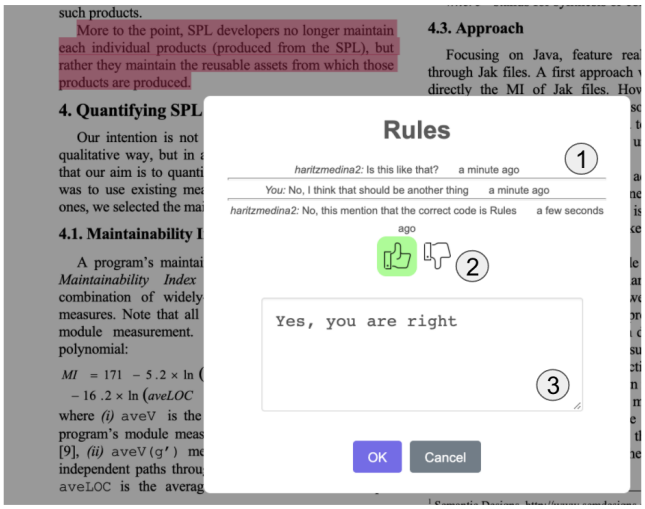


Figure 3.12: Cross-checking discussion and decision taking. (1) Discussion log where reviewers have a conversation about the classification over the annotation context (i.e., the paper). (2) Validation buttons allow validating or invalidating a classification. (3) Comment text input to continue the discussion or provide a reason or note for validation.

that led each of the reviewers to reach a final agreement. For the reconciliation phase (i.e., discussion to reach a classification agreement), *Highlight&Go* allows extractors and the manager of the secondary study to follow a discussion using assessing-like annotations (see Fig. 3.8) and the manager can mark the classification annotation as validated or invalidated (see Fig. 3.12). All these annotations are also translated automatically to the spreadsheet as we will see next.

Transparent storage of coding event logs in the spreadsheet

This mechanism is closely related and is a consequence of what users have done in the previous three writing mechanisms. Persistence in *Highlight&Go*, is an automatic process that populates a created spreadsheet with all events carried out by all reviewers in the data extraction activity of the literature review. Web annotations, as mentioned in Section 3.4, can work as event logs for all decisions taken during data extraction activities: codebook definition, primary studies' metadata, annotation process, responses between reviewers, and cross-checking discussion and resolution. Primary study metadata is serialized in a spreadsheet following three ontologies: *oa* (the ontology for W3C Web Annotation), *bibo* (the ontology for bibliographic references), and *dc* (an ontology for metadata). These annotations are serialized in spreadsheet rows, one annotation per row, and each column per annotation attribute (see the example of the codebook definition in Fig. 3.13).

	A	B	C	D	L	M	N
1	oa:Annotation	dcterms:created	dcterms:creator	oa:hasBody	oa:motivatedBy	schema:potentialAction	multivalued
2	YuoyRLReoKPAo1IVXJq2M2	2021-10-18T08:01:50.528Z	haritzmedina	Asset type	slr:codebookDevelopment	schema:CreateAction	TRUE
3	2Rc0hTaZg9dslr7rS83pXc	2021-10-18T08:01:56.990Z	haritzmedina	Code assets	slr:codebookDevelopment	schema:CreateAction	FALSE
4	_f5jQ9mkrj6-xvqwePoXFQ	2021-10-18T08:02:02.745Z	haritzmedina	Products	slr:codebookDevelopment	schema:CreateAction	FALSE
5	fJnCFdYqeW3cly7gWqzYJr	2021-10-18T08:02:09.029Z	haritzmedina	SPL architecture	slr:codebookDevelopment	schema:CreateAction	FALSE
6	5xkQLqiTvcnFWatkf2wOAg	2021-10-18T08:02:16.089Z	haritzmedina	Variability model	slr:codebookDevelopment	schema:CreateAction	FALSE
7	P8DoAe_Z6cF1nnGryUHDFq	2021-10-18T08:02:22.479Z	haritzmedina	Research type	slr:codebookDevelopment	schema:CreateAction	TRUE
8	S99E5ayEE7l99NN_8UpJZL	2021-10-18T08:02:30.166Z	haritzmedina	Conceptual prop	slr:codebookDevelopment	schema:CreateAction	FALSE
9	MahO7UR9lIVoPBDSTC5Iz_	2021-10-18T08:02:38.262Z	haritzmedina	Evaluation resea	slr:codebookDevelopment	schema:CreateAction	FALSE
10	nkT3JaTiDonWs0HeEc4VBr	2021-10-18T08:02:43.855Z	haritzmedina	Experience pape	slr:codebookDevelopment	schema:CreateAction	FALSE
11	bcucojlU3GLcXOJnbND25B	2021-10-18T08:02:50.997Z	haritzmedina	Solution proposa	slr:codebookDevelopment	schema:CreateAction	FALSE
12	27CNp77mTDYR0yvyAr3Hxo	2021-10-18T08:03:00.240Z	haritzmedina	Validation resea	slr:codebookDevelopment	schema:CreateAction	FALSE
13	4hZaJQDBymshCn2tvgY0yi	2021-10-18T08:03:17.797Z	haritzmedina	Product-derivati	slr:codebookDevelopment	schema:CreateAction	TRUE
14	0XfkYNBt9lqnA_uCS5NJK1	2021-10-18T08:05:24.130Z	haritzmedina	Annotation-base	slr:codebookDevelopment	schema:CreateAction	FALSE
15	6zIlmvLvu2yKXTakH2D80l	2021-10-18T08:05:32.432Z	haritzmedina	Composition-bas	slr:codebookDevelopment	schema:CreateAction	FALSE
16	OkZ87vH67J8Zz35ZMfe1dp	2021-10-18T08:05:43.699Z	haritzmedina	Evolution activity	slr:codebookDevelopment	schema:CreateAction	TRUE

Figure 3.13: Serialized annotation for codebook development in Google Sheets (partial view). Each of the annotation attributes is in one column, which may vary in different annotation types.

All these annotations are automatically transcribed to five different spreadsheets, one per type of annotation described in Section 3.4 (*classifying*, *codebookDevelopment*, *categorization*, and *assessing*) and another for primary study metadata. These logs act as a database that reading mechanisms will consume to provide visualizations automatically in the same spreadsheet. We will see them in the next section.

3.5.2 Read: Spreadsheets

Highlight&Go, as mentioned above, combines a browser extension and a Google Sheets script. This script is embedded in all spreadsheets created by *Highlight&Go* browser extension. While the browser extension acts as a writer, the spreadsheet acts as a reader of created annotations. To facilitate traceability, consistency, completeness, and observability, Google Sheets and its embedded script render annotations in "a la spreadsheet" view.

Tabular classification for coding overview in spreadsheet

The common convention to report the results of secondary studies is the use of a matrix. One of the axes accounts for the papers analyzed, while the other accounts for the themes that help to respond to the research questions. Cells include extracted data (PS's quote) or classification decisions made for the pair <paper, theme> (a code). *Highlight&Go* generates from stored annotation event logs (see spreadsheet in Fig. 3.13) a matrix with papers and themes. This accounts for familiarity, as follows the same practice of traditional data extraction. In the example in Fig. 3.14 the previously annotated SPLEMMA paper in row 4 is presented, where for the *Asset Type* theme the paper is classified as *Code assets* and *SPL architecture*. As *Asset type* is a multivalued theme, papers can be classified with more than one code for the corresponding theme. However, the *Evolution Activity* theme for the SPLEMMA paper has a single column, since the theme is mono-valued (i.e., the paper can only address a single

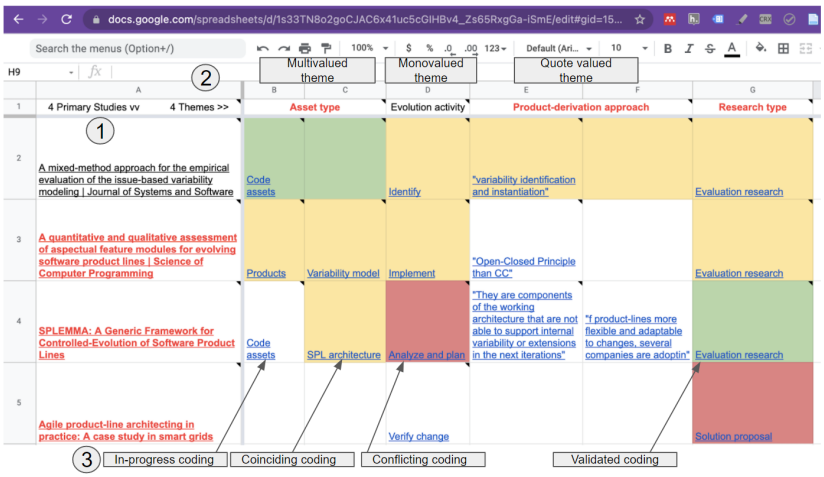


Figure 3.14: *Highlight&Go* annotation production: extending *Google Sheets* with query functions upon annotation logs in Fig. 3.13. Cells point to web resources, ready to be navigated upon. To see this example in action, move to the following *Google Sheet* https://rebrand.ly/hGoSheet21.

evolution activity) and in this case, is classified as *Analyze and plan*. Another possibility is that the theme has no code. For instance, reviewers capture data (i.e., annotate evidence with the theme) but do not have a code to classify with, so the annotated quote is shown that will help for further analysis (e.g., conduct a thematic analysis).

Thus, the spreadsheet provides the result of the data extraction activity. However, it is very general information, which is not useful during data extraction or even traces back evidence in analysis and reporting. *Highlight&Go*'s spreadsheet not only holds the result (i.e., the cell value) but also provides additional information to facilitate consensus and information gradation to ease traceability. Let us introduce the consensus mechanism, the cell color semaphore.

Cell color semaphore

The cell background color supports up to four different colors, denoting the review progress for the pair <paper, theme> (see Fig. 3.14):

- White color denotes that no problems have been detected for the <paper, theme> at hand, but that is pending to be reviewed by a second extractor.
- Yellow color denotes that the pair <paper, theme> at hand is classified by more than one extractor independently without any kind of conflict.
- Red background denotes a conflict (i.e., a lack of consistency) that arose

during data extraction for a pair <paper, theme>. Three types of conflict can arise:

  - Intra-extractor inconsistency arises for a single extractor, where a theme can only be coded with a single code and the reviewer has classified it with more than one code.
  - Inter-extractor inconsistency arises when different extractors classify with different codes the same mono-valued theme in the same paper.
  - Inter-validator inconsistency arises when a checker has validated more than one code in a mono-valued theme, as it can only take one possible value, but more than one is validated.

- Green background denotes that a checker or the person responsible for the review has validated a classification.

The cell color mechanism can help during data extraction to denote pending consensus problems to be solved, but also themes that are not accordingly reviewed by more than one extractor, which may result in biased coding. However, going through each of the cells in a big spreadsheet with lots of papers and themes is not scalable. To solve this and provide information at the theme or primary study level is also necessary. This is where the themes' and primary studies' font color semaphore comes into play.

Theme and Primary Study level semaphore

Reviewing progress status is important to make reviewers, but especially the manager of the secondary study, aware of the current state of data extraction activity. Some of the questions a person in charge of the secondary study (or an extractor itself) can make during the data extraction activity are: "are there any disagreements between reviewers for the same paper?", "how many themes are not classified?", "how many papers are pending to be peer-reviewed?", "how many papers are already checked?". To help in the awareness of what is done (classification is done and validated) and what is pending (inconsistent or incomplete coding) during data extraction activity *Highlight&Go* has enhanced spreadsheets using cell coloring (see Fig. 3.14).

Incompleteness is defined as pending or lack of coding in papers or themes:

- Non-peer-reviewed incompleteness arises when a paper has been classified by a single reviewer. This keeps the cell on a white background and it turns yellow when a second reviewer has conducted the classification with the same selection of codes or red if any previously described conflict arises.
- Non-coded paper incompleteness arises when a paper is not classified in all themes. This is denoted with red and bold text in the cell where the paper's title is in the first column.
- Non-coded theme incompleteness arises when there exist papers pending to be classified by this theme, and the theme name in the first row becomes in red and bold.

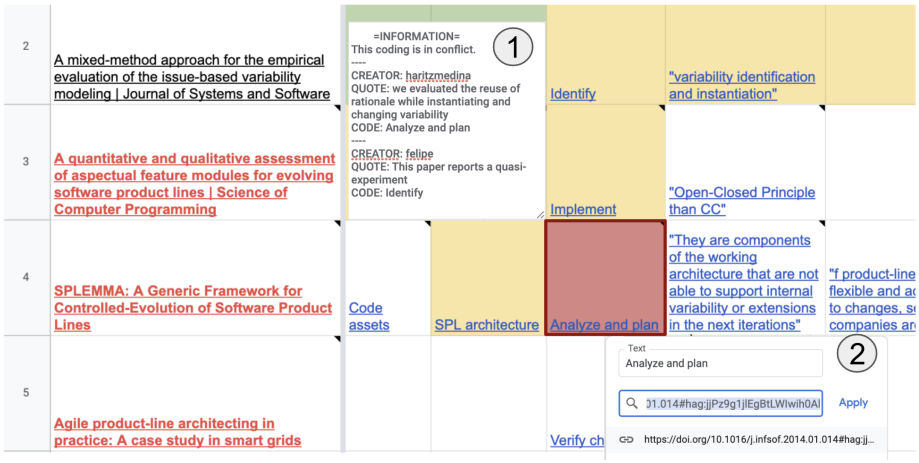


Figure 3.15: Decision information is gradually shown. (1) hovering over a cell with a classification decision shows who has classified, on what evidence is based and provides additional comments. (2) adds to each cell a hyperlink to move to the evidence shown in the context (i.e., the annotated quote in the paper on the web).

The color semaphore acts as an awareness mechanism to facilitate looking up problems that must be tackled by the reviewing team. However, further explanations should be provided to know exactly what the problem is and how it should be solved. To avoid overwhelming users with too much information, this information is shown gradually using Google Sheets notes, and hyperlinks.

Hyperlinks linking coding decisions to context

The information gradation mechanism is used to gradually display the information that reviewers need during data extraction in the spreadsheet. The spreadsheet shows the classification of the papers within the codebook. The cell color semaphore denotes where reviewers must focus to finish the data extraction activity with success.

However, the matrix is a single summary of classification results and current status but does not provide information on how to solve the incompleteness or which ones are inconsistencies in the coded data. There, *Highlight&Go* enhances the spreadsheet by reusing already existing two mechanisms in Google Sheets: notes and hyperlinks.

Google Sheets permits the attachment of textual notes to the cells. This is used as a mechanism to append extra information related to the cell that can be seen when the user hovers the mouse cursor over it (see Fig. 3.15). Notes can act as a second-level information gradation. *Highlight&Go* uses these notes to provide extra information about the coding process or problems that arise during coding activities.

Hyperlinks point to web resources. Therefore, the W3C recommendation does not only provide a data model for annotation description but conceptualizes annotations as web resources, and hence, as URL addressable. *Highlight&Go* enriches spreadsheet cells, where classification decisions are shown, adding the URL to the web annotation that sustains that decision in the context (in the classified primary study). This link opens the primary study at the exact position where the annotation was taken. There, it is possible to analyze the classification taken, navigate through all annotations that sustain the classification decisions, and check and reach an agreement as explained in Section 3.5.1.

3.6 Intervention

ADR emphasizes continuous evaluation, unlike a separate stage of the research process that follows building. While the researcher may guide the initial design, the artifact emerges through the interaction between design and use [SHP+11]. This allows researchers as well as organizational stakeholders, to shape the artifact over the research lifecycle.

In the research project, there was an interplay between the development of theoretical contributions and the development of the data extraction tool. The theoretical contributions were not only informed by the existing theory, but they were also highly influenced by empirical evidence collected from the development and evaluation of the data extraction tool. In parallel, the design of the data extraction tool was based on the emergence of contributions to theory. Consequently, there was a mutual influence between the emergent theoretical contributions and the development of the data extraction tool. Sein et al. do not operationalize how this researcher-practitioner relationship is to be accomplished. In the first cycle, the data extraction process was monitored by conducting informal interviews with practitioners to provide troubleshooting, gather the main problems when interacting with the tool and how its design can be improved. This process resulted in six main improvements from the initial release of *Highlight&Go*.

For the second cycle, we were looking for how extractors interact on their own, without the help or influence of the tool developers. To this end, we resorted to Daily Diaries. Daily Diary Methodology is a set of assessment methods that allow researchers to study the experiences, behavior, and circumstances of individuals in natural settings, in or close to real-time, and on repeated measurement occasions over a defined period (ranging from a few days to months) [SKGA17]. Daily diary entries permit a prompt record of the experience (episodic memory) without the risk that these experiences fade away. When individuals are asked to summarize their experience over a specific time interval, the summarized rating may be unduly influenced by the most recent and the most intense moments of the experience [K+99].

A diary, according to Lazar et al. [LFH17], enables the collection of data over time, and the participants are in charge of selecting the location and times

for the records. This makes it easier for individuals to record the specifics of events, as remembering everything by the conclusion of the data collection time would be difficult. A diary study "minimizes the influence of observers on participants" [CM05]. Participants have more autonomy in registering their responses because the researchers are not there at the time of the occurrence. Furthermore, since the participant is not interacting with the researcher during the diary annotations, the participant is more likely to act naturally, which may assist the researcher to gain a better understanding of the events that were most meaningful to the participants.

By using the daily diary methodology, researchers can adequately address research questions on within-persons processes — that is, processes that unfold within individuals over time. This makes sense to alleviate the novelty bias as well as for some product features that might be time-related. For instance, access to previously classified pieces of evidence makes sense as long as previously analyzed primary studies exist. The utility of such capabilities can be better assessed once distinct primary studies have been reviewed. The next paragraphs instantiate the steps of the daily diary methodology [Sal16] for our case. *Highlight&Go* was put to the test by agreement with three researchers. The test was naturalistic: real tasks (i.e., actual papers to be classified and data coded), real users (i.e., researchers), and real setting (i.e., results are uploaded and disseminated to real conferences and journals).

Diary Log Structure. Guidelines for the diary include questions to be short, easy-to-understand questions that go beyond yes/no, and a reduced number of questions to avoid tiring the participants. Questions were proposed to capture the different states of data extraction. This included:

- Explain your experience in handling codebook-related issues (e.g., codebook definition, codebook update, highlighter realization).
- Explain how smooth was your coding (i.e., highlighting) experience.
- Explain how smooth your commenting experience was.
- Explain how smooth was your consensus reaching experience (e.g., viewing consensus, browsing to evidence, annotating agreement) if there was any.
- Explain how seamlessly you found the interplay between *Google Sheets* and *Highlight&Go*.

Planning and Preparation. The researchers introduced the participants to the data extraction activity, how the data can be traceable [GF17], and the main properties of web annotations to help in the labor. An initial session of one hour was held explaining the main functionalities of *Highlight&Go* and an example based on a real review of the literature. The fact of having a sort of standardized example and main mechanisms of the tool was aimed at facilitating the focus while reporting on the pros and cons of the intervention. Each participant tested the tool in the following days before conducting data extraction in a real setting (the literature review they conducted).

Logging period. In-situ logging was used to collect data. Participants were asked to log information about the data extraction activity right after each session with the tool. It should be noted that a review of the literature is required for distinct tool sessions. We resorted to a Google Form to realize the log. Data collected include participants' ID, day, starting and ending time, and 5 textual boxes to answer the 5 questions that account for the mechanisms of the annotation tool and the interplay between *Highlight&Go* and Google Sheets.

Post-study interview. The ADR team periodically (approximately every two weeks) monitored diary studies, analyzing the information provided by each participant. We asked probing questions to uncover specific details that were obscure in the logging report.

During the diary study, the ADR team noticed that participants (after some sessions) were providing less information in the diary logs. When the ADR team inquired about this concern, the participants gave three main explanations for the absence of feedback:

- Participants had some doubts about what should be reported and what not.
- They had some doubts about knowing where (in which question) they should report specific experiences that affect more than one mechanism.
- They reported that sometimes everything worked well or as expected and that they did not have additional feedback to provide.

At the end of the diary study, a confirmatory focus group was held to gather general impressions of the tool during the data extraction activity. The next section reports the general benefits introduced after the diary study and the evaluation of the participants.

3.7 Evaluation

The purpose of the evaluation activity is to analyze the knowledge collected from the participating extractors to develop a proposal for an efficient, effective, and portable data extraction utility.

3.7.1 Impact on the tool

This subsection summarizes the main insights provided by practitioners during the first ADR cycle that impacted the current version of the tool. As the extension was already publicly available in the Chrome Web Store and was presented as a prototype at the EASE [DMA19] and JISBD [MDA18] conferences, third-party users' comments were also collected to improve the usability of the tool.

Move from annotation systems storage to spreadsheets. In the first release of *Highlight&Go* the annotation storage used was *Hypothes.is*. The change from *Hypothes.is* storage to Google Sheets has a two-fold goal. On the

one hand, users were burdened by the requirements to start data extraction, as initial versions of *Highlight&Go* required the use of a *Hypothes.is* account (to store annotations following W3C) and a Google account (to create spreadsheet visualizations). Even if the interplay between the two realms was valued as positive and smooth, the requirements to head-start data extraction were considered high by early adopters of the tool. On the other hand, the reuse of annotations in a different way by researchers for further analysis is hindered in *Hypothes.is*. Retrieving annotations from *Hypothes.is* requires access to an API, while annotations stored in spreadsheets do not require a technical background to perform analysis. Due to these two facts, we moved the implementation of annotation storage from *Hypothes.is* to Google Sheets, requiring users only to sign up for one service to start using the tool.

Inclusion of all the quotes in the spreadsheet. For quote-valued themes (see Fig. 3.14) initial design of *Highlight&Go*, to facilitate scalability, only showed one of the annotated quotes in the paper, requiring extractors to click on the link and see the rest of annotated evidence in the paper using the overview & detail mechanism (see Section 3.5.1. However, this hindered the overview of all the quotes and reuse of all the quotes in a third-party tool (i.e., exporting the generated table by *Highlight&Go* to statistical software R) in the analysis and synthesis step of the literature review.

Mark in gray instead of removing annotations attached to removed code/theme. In the early releases of *Highlight&Go*, the annotations were closely attached to a theme or code, which means that if there were changes in the codebook, such as removing a theme, all the annotations made or classified with that theme were also removed. This forces reviewers to re-read and reclassify the paper, but it can be interesting to keep the previously extracted evidence to help in that reclassification of the paper. To this end, annotations are not automatically removed but are shown in gray to help reviewers retrieve previously highlighted ground data.

Improvements in metadata discoverability and retrieving. Participants have used *Highlight&Go* in other services apart from those supported by default (IEEE, ACM, Springer, ScienceDirect), which makes some websites not correctly retrieve paper information (e.g., title). In the beginning, the spreadsheet view was included at the end paper's metadata, which was desegregated to another sheet (in a more familiar format) to let users modify/correct manually any metadata that *Highlight&Go* could not capture on its own.

Inclusion of all the quotes that sustain a decision in the spreadsheet cell. At the very beginning, web engineers developing *Highlight&Go* when providing information gradation using spreadsheet notes, included all the annotated quotes that sustain the classification. In later releases of *Highlight&Go*, quotes were removed from the note to facilitate reading and scalability. However, practitioners pointed out that, even if sometimes can be less scalable, in most cases it is useful to have all evidence in the note to avoid reviewers switching to the reading realm when it is not necessary. Therefore, this feature was released back again.

Reclassification mechanism. One of the most demanded features to be

included was the possibility to reclassify annotations (e.g., change the code or the theme that an annotation pertains to). Even if the codebook is defined in the review planning phase, there can always arise (agreed) changes to the codebook (e.g., the inclusion of new codes in a theme or split of one theme into two different ones). The previous versions of *Highlight&Go* required users to delete an annotation and create a new one to reclassify an annotated evidence. Participants suggested providing a specific feature that allowed them to select the annotation they wanted to modify and choose a code or theme. New releases of *Highlight&Go* will include this feature to improve the user experience.

3.7.2 Impact on the data extraction process

This work presents *Highlight&Go* as an intervention for efficient and effective data extraction. According to the inner-outer model presented in Fig. 3.2, we want to validate to what extent *Highlight&Go* mechanisms facilitate efficiency, effectiveness, and portability. To this end, we have defined a qualitative and quantitative evaluation:

- A qualitative evaluation to gather impressions about *Highlight&Go*'s utility for conducting data extraction in secondary studies, to what extent it has impacted the data extraction process, to what extent it facilitates further analysis and captures future directions or new functionalities or scenarios where the tool can be used. To this end, we resorted to diary studies during the intervention (presented in Section 3.6) and a confirmatory focus group after the intervention 3.7.3.

- A quantitative evaluation (i.e., a comparison benchmark) that compares the effort researchers need, in terms of clicks, to conduct the tasks required in data extraction by doing them manually (based on Garousi's approach [GF17]) and with the help of *Highlight&Go*.

3.7.3 Qualitative evaluation: confirmatory focus group

The evaluations conducted in the first cycle facilitated the improvement in the design of the solution. However, it has not been measured by other researchers to what extent *Highlight&Go* is useful to conduct data extraction in a literature review. We conducted a focus group with three researchers from two different institutions that have been using the tool in a real setting. Next, we describe the research design and present the main results from the analysis of the focus group.

Research design

Objectives of the study. The qualitative evaluation consists of two main parts, the diary study, and the confirmatory focus group. The purpose of the diary study was to cover aspects that arose during the use of the tool. This

Table 3.2: *Highlight&Go*'s evaluation participants (i.e., researchers) characterization.

Researcher ID	R1	R2	R3
Years as active researcher	3	2	1
Ph.D. area	Computer Science	Computer Science	Information tech. and mobile networks
Ph.D. topic	Software Product Lines	Design Knowledge accumulation and evolution	Privacy Preserving Analytics
# of published papers	6	2	0

is complemented by the confirmatory focus group, which has been used to recap and discuss the main benefits and pitfalls of using *Highlight&Go* for data extraction. To this end, the purpose of the focus group is:

> to discuss *Highlight&Go*'s utility for conducting data extraction in secondary studies, to what extent it has impacted the data extraction process, to what extent it facilitates further analysis, and to capture future directions or new functionalities or scenarios where the tool can be used

To this end, we defined the focus group protocol with nine sections in mind from most general to specific questions (see Appendix A).

Participants identification. Identification of participants is perhaps the most critical step since the technique is largely based on group dynamics and synergistic relationships among participants to generate data [PT06]. Most of them recommend aiming for homogeneity within each group to capitalize on people's shared experiences. We selected three participants for the qualitative evaluation who had planned to conduct a secondary study, had easy access to them, and were ready to evaluate the tool *Highlight&Go* for data extraction. This resulted in three participants, two Ph.D. students part of our research group, and a third one, a Ph.D. student in the Tecnalia research center. The minimum requirement for selection is that they have been working with *Highlight&Go* for more than two months to conduct data extraction of a secondary study, where they extracted data from more than 30 primary studies, and that they had at least one year of experience in research. Table 3.2 summarizes the experience of the participants and Table 3.3 the context in which *Highlight&Go* has been evaluated.

Identify the moderator. Due to the open-ended nature of focus groups, moderation can be complex. For this study, the moderator was one of the

Table 3.3: Literature reviews conducted using *Highlight&Go*. LR1 corresponds to the literature review conducted by the researcher R1 characterized in Table 3.2.

Literature review ID	LR1	LR2	LR3
Review type	SMS	SLR	SLR
Review purpose	Depict visualizations in variant-rich systems	Identification of problems research software reuse	Analysis of secure multiparty computation approaches
Guidelines used	[PFMM08, KBB15]	[KBB15]	[KBB15]
# of participant researchers	4	2	2
# extractors	2	1	1
DE strategy used	100% independent extraction. Solve mismatches	Lone researcher & supervisor's sample review	Lone researcher & supervisor's sample review
# of primary studies	42	73	32
# of themes & codes	13 & 42	11 & 42	13 & 108

authors, while another author played the role of the observer, who recorded supplementary data during the discussion and provided support to the main moderator.

Location. Due to the availability and location of participants in two different cities, we resorted to an online focus group [KD13]. Online focus groups are not a different type of focus group discussions, but they are applied to the online environment. The session was held using the Webex conferencing application.

Data collection

The moderator contacted the three participants after using *Highlight&Go* for evaluation purposes in a real setting. Permission was obtained to perform the evaluation. During the focus group, the moderator acted as a facilitator, posing questions and prompting follow-up questions, encouraging the participants to elaborate on certain points and offer additional comments. It lasted for one hour and 45 minutes. In Appendix A a short guide is presented followed by the moderator with the main points and questions presented during the session. In addition to the notes taken by the observer, the session's audio was recorded for analysis.

Analysis

The session's audio was recorded and transcribed using Sonix[8] and manually reviewed. A grounded theory approach has been followed and transcriptions

[8] https://sonix.ai/

Table 3.4: Participants' perceptions of the utility of *Highlight&Go*'s mechanisms.

Mechanisms	R1	R2	R3	Total
Transparent extraction of coding to spreadsheet	7	8	8	23
Codebook-based color-coded highlighter	8	6	7	21
Hyperlinks in cells linking coding to context	6	7	3	16
Tabular classification view in Google Sheet	5	5	6	16
Overview & navigation over primary study coding	3	4	2	9
Theme and primary study level semaphore	2	1	5	8
Transparent extraction of metadata	4	3	1	8
Cell color semaphore	1	2	4	7

were coded independently by two researchers [CS90]. For the analysis, we have used *Highlight&Go* itself [9]. We began by creating an initial template with high-order codes and some lower-order codes that focused on aspects of the data quality metrics.

Results and reporting

This section introduces the main results from the diary study and the confirmatory focus group conducted at the end.

***Highlight&Go*'s utility for data extraction**. One of the main objectives of the evaluation was to discuss to what extent *Highlight&Go* is useful for data extraction. Participants were asked to rank *Highlight&Go*'s mechanisms from 8 (the most useful) to 1 (the least useful). Table 3.4 shows the results.

At the bottom of the ranking is placed the cell color semaphore and in 6th place is the theme and primary study cell semaphore. The little success of both facilities was related to the followed strategy by participants. After asking about this matter, one of them mentioned: "the consensus using colored cells can be useful, but in my case, I was the only extractor so it was useful to spot doubts. Sometimes, I classified a mono-valued theme with two codes because I was in doubt, the cell turned red, and then I was able to review. However, since no one else extracts the entire paper, green or yellow colors were not used". Another researcher mentioned that: "I collaborated with an extractor from another research center that prefers to use another tool for data extraction, so the consensus step was done outside *Highlight&Go*". Looking at the transparent extraction of metadata, no one disagreed that "automation is good, but in my case, I have extracted the metadata of papers before from the results of search strings, so I did not find it very useful". The overview in primary studies was reported as quite useful: "is one of the features that helped me most, as I was able to navigate through the evidence easily, especially in long documents".

[9] *Highlight&Go* was used to extract quotes directly to Google Sheet where a further analysis was conducted using sheets query facilities.

On the other side of the coin, participants placed the tabular classification views on Google Sheets in 4th position. Some of them think that usability should be improved: "classification was presented as expected, but usability should be improved, sometimes it made some columns too wide automatically to keep the whole extracted quote on the screen, and required me to change manually the size of columns and rows". However, all of them agree that the visualization "is quite familiar and presented in the same way as most of the SLRs do, with two dimensions, one for papers and the other for themes". In the same way, hyperlinks to trace back to evidence were positively valued, but during the discussion arose that "maybe it is more useful for literature reviews where a type of thematic analysis should be done, as reviewing quotes and context is more used in those cases". It should also be noted that "it is more useful at the beginning of the literature review when you have more doubts about classifying a paper, you can easily navigate to previous ones to make a consistent classification". In the same vein, the most experienced Ph.D. student addressed that "for me it was more useful to check the description of a theme or code in the sidebar, where I placed an example on how I have to classify. However, I indeed missed a functionality that allowed me to go directly or move to previous papers with the same classification instead of requiring me to move to the spreadsheet, navigate through the paper and click the link".

The top two mechanisms were the color-coded highlighter and transparent extraction of data to the spreadsheet. Regarding the first, one of the uses that the Ph.D. students gave to the color-coded highlighter was that "I created a miscellaneous theme to extract interesting stuff that maybe I might need in the future, but at the time of classification, I did not know how to classify it. This helped me a lot, but the reclassification of annotations was not easy at all in *Highlight&Go*, so I decided to delete and create again the same annotation with the new coding". All three agree that color-coding was useful to easily spot quotes and to always classify the paper with the same code, but "in some cases all evidence was in the same paragraph, and the highlights were difficult to spot. To solve this, I would suggest the use of a filtering mechanism based on themes". The researcher who had to share his classification with the other researcher who did not use the tool mentioned "when we merged both classifications into a new spreadsheet, I realized that the other researcher did not always code the exact name in the corresponding cell. Sometimes he capitalizes the name of the theme, others the name was different, singular and plural, etc. requiring data scrubbing before analysis. As I have used *Highlight&Go*, there were no transcription errors and the processing of the resultant spreadsheet was faster". Aligned with the color-coded highlighter, one of the researchers remarked on the utility of comments in more than one diary study log: "Commenting is a functionality that the more I use it, the more useful I find it. I use it to explain why I have decided to classify a text fragment with a concrete code and in our weekly meetings those comments help me share my concerns with my colleagues". Another participant added: "comments reuse saves a bit of time as I used the comments to provide extra explanations about my classification decisions, and this makes me more consistent when memoing those reasons".

Impact on data extraction process and possible improvements. Even if for all three cases it was the first time they had conducted a literature review using a systematic process and results, they found the tool easy to use. At the very beginning, they requested some help to let them know how to use some features or to make them aware of some features that are not easy to find, like the navigation among highlights using the sidebar, but they did not have learnability issues. In the same way, they did not find it very convenient at the beginning to access and annotate the papers in the digital libraries. Some of them preferred to download and annotate the documents offline, which is possible in *Highlight&Go*, but depending on the PDF metadata discovery (presented in Sec. 3.5.1), sometimes it is not possible to trace back to evidence from the spreadsheet using the links. However, they feel beneficial as a tracing back mechanism and get used quickly to annotate directly on the web or the PDF version on the web.

All the researchers agreed that *Highlight&Go* have improved their process, or at least facilitated them to conduct the DE more effectively and efficiently. Three researchers conducted DE, in which two of them were merely required to extract quantitative data or classify papers based on predefined themes, where *Highlight&Go* adapted well to their way of work. However, the second researcher mentioned that sometimes feels that "*Highlight&Go* is a good tool for extracting quotes, but lacks mechanisms to facilitate thematic analysis, like splitting or joining themes and codes, and it jeopardizes my way of work sometimes". To solve those problems, he configured some queries over *Google Sheets* to retrieve the data, which he needed to conduct the thematic analysis, but he proposed the use of mind maps or a kind of integration with a mind mapping tool as a better way to conduct thematic analysis.

In conclusion, participants were able to successfully conduct their literature reviews using *Highlight&Go*. They appreciate most the integration between the reading realm and the spreadsheet in both directions: highlighting paragraphs to automatically populate the spreadsheet, and from the spreadsheet to revisit the annotated paper. The links to move from the spreadsheet to annotations not only improved the data extraction phase but also the analysis and review report phases. One of the most important issues is the lack of support for thematic analysis, which should be improved in the next releases of the tool. Next, we move to the quantitative comparison made in terms of performance between manually conducting the data extraction and using *Highlight&Go*.

3.7.4 Quantitative evaluation: comparison with Garousi's standalone spreadsheets

Garousi et al. [GF17] propose the use of spreadsheets to systematically extract the required data from primary studies and accurately record the information researchers need to answer the questions of a literature review more effectively and efficiently. The method is to record in the spreadsheet taking decisions and color-coded highlighting in primary studies (i.e., as same as *Highlight&Go* does, one code for each theme). In this way, traceability is supported by reducing

Table 3.5: Comparison between manual data extraction following Garousi's approach and Highlight&Go. The comparison has been done in 5 different settings (the example provided by Garousi and the four literature reviews in our group presented in Sec. 3.2.1).

Review	GF17	DA15	MD16	PD19	AMD21
P = # PSs	78	42	107	30	66
T= # Themes	5	1	4	3	5
C= # Codes	29	6	18	15	14
M= # Metadata	3	3	9	1	6
SR total actions manual approach	**14,075**	**1,772**	**12,863**	**2,839**	**6,158**
Setting up = 1 + P + T + C	113	50	130	49	86
Collecting a PS's metadata = 2 + (4 * M)	14	14	38	6	26
Coding a PS = 2 + (6 * C)	176	38	110	92	86
SR total actions H&G approach:	**7,546**	**1,120**	**6,722**	**1,662**	**3,376**
Setting up = 4 * (T + C)	136	28	88	72	76
Collecting a PS's metadata = 6	6	6	6	6	6
Coding a PS = 2 + (3 * C)	89	20	56	47	44
% of efficiency improve in Highlight&Go vs manual	**46.39**	**36.79**	**27.74**	**41.46**	**45.18**

errors (e.g., incorrectly classifying a primary study). However, this process is done manually, requiring extractors to copy and paste classification decisions, evidence from the paper, etc. *Highlight&Go* goes a step further automatizing part of the actions that extractors have to do manually in Garousi's approach. We have conducted a quantitative comparison of the minimum number of actions (clicks, copy-pasting, typing, moving from the reading realm to the spreadsheet all the time, etc.) required to capture traceable data. Table 3.5 shows the results of such comparison for five different literature review settings in three main steps of data extraction: setup DE form (i.e., spreadsheet columns in manual approach and codebook in *Highlight&Go* approach), capture metadata (e.g., year, title,...) and coding a paper (i.e., color annotation + filling the spreadsheet). The DE strategy followed to measure in both approaches is a single quote is captured to decide the classification by a single reviewer. In all the steps and all the secondary study settings that have been calculated, *Highlight&Go* is between 40% and 50% (depending on the review setting) more efficient in terms of the number of actions required by the extractors to conduct DE. The full explanation of the interactions measured for each of them can be found in https://rebrand.ly/comparisonWithGarousi.

3.7.5 Threats to validity

Construct validity refers to the degree of accuracy with which the variables defined in the study measure the construct of interest. Here, we resort to a combination of quantitative and qualitative evaluation [Kit96]. For quantitative evaluation, we account for the number of interactions (e.g., clicks) that a researcher needs to define the extraction spreadsheet, capture metadata, and code a primary study. However, more steps are conducted during DE and, for example, the efficiency results may vary depending on the DE strategy followed. Looking at the qualitative evaluation, diary studies and focus groups can pro-

vide rich information as preliminary or larger-scale research. However, they are vulnerable to manipulation by a facilitator and trust between participants as well as between participants and the moderator, to avoid power talkers taking a particular view, where then others are likely to agree even if they disagree [BE11]. To validate the findings of the focus group, we emailed all participants inquiring about their agreement.

Internal validity looks into the extent implemented mechanisms are the ultimate cause of facilitating the efficiency and effectiveness of data extraction for researchers. From this perspective, other factors besides *Highlight&Go* might influence the results. The diary study results rely on self-reported data. This is counter-measured by periodic monitoring and meetings to ask them to solve misunderstandings or incomplete data. However, the results may lack completeness, as participants are new to this evaluation methodology and have doubts about what should be reported or not raised during the evaluation. To counterpart the lack of completeness in the results gathered, a confirmatory focus group was conducted after the evaluation period.

External validity tackles the representativeness of the study and the ability to generalize the conclusions beyond the study itself. In this work, a qualitative evaluation has been conducted in a real setting with the intensive use of the tool for at least more than two months by three Ph.D. students. At this point, the number of participants is low and only novice researchers from two different institutions were invited, which may compromise the results. However, Ph.D. students are also one of the most interested stakeholders. To enter a research area and learn about it or find research questions to address during Ph.D., it can be quite beneficial to start with a literature review (i.e., State of the Art or Systematic Mapping Study) [PB14]. However, more evaluations are required to improve the soundness of this study with more experienced researchers.

3.8 Formalization of Learning

As a sibling of Action Research, ADR intervenes and improves a specific setting through a cycle of making changes, observing the resulting situation, and making further changes. The researcher is "experimenting" by making adjustments and observing the effects of those adjustments. This improves ecological validity but hinders generalizability (i.e., findings are limited in their transferability to other settings as interventions are likely to be dependent on the specific organizational context, e.g., the use of Google Sheets) and precision (the natural study setting is realistic but subject to confounding factors that limit the precision of measurement). Hence, ADR should include a reflection of the extent of these threats. Sein et al. [SHP+11] suggest three levels for this conceptual move:

- generalization of the problem instance, i.e., to what extent is "data extraction efficiency and effectiveness" a *problem* for other researchers apart from our university department;

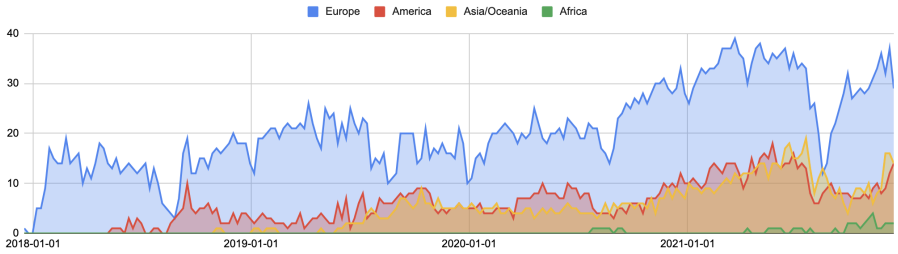


Figure 3.16: *Highlight&Go* user base from its release at the end 2017 in Chrome Web Store distributed by region (source: Google Chrome Store).

- generalization of the solution instance, i.e., to what extent is *Highlight&Go* a *solution* for the above problem; and
- derivation of design principles, i.e., what sort of design knowledge can be distilled from the *Highlight&Go* experience that might inform other interventions.

3.8.1 Generalization of the Problem Instance

It is worth noting that the problem of tool support has been identified in multiple studies [HCHAZ16, GF17]. Automation is a must to reduce the effort required in the entire literature review process, and many efforts have been made to provide the tool support to build protocols, plan the review, conduct search, select studies, or generate graphs and reports. However, automation in DE is more limited and the large number of manual steps to be conducted makes the process more time-consuming [AZCHH17].

On the other side, literature reviews produced are willing to share their laboriously obtained data. Literature review consumers demand access to these data to capitalize, review, and reuse the data [NS18]. Yet, the lack of portability is making these wishes elusive [AZCHH17]. Most QDAS tools provide proprietary formats, which hinder data validation and repeatability of the study by third parties. Additionally, the lack of portability does not facilitate moving from one tool to another. As there is not a "one-size-fits-all" literature review tool giving support to all steps in a literature review, the problem is relevant in academia.

3.8.2 Generalization of the Solution Instance

We now address the extent *Highlight&Go* might be a *solution* to conduct portable, efficient, and effective data extraction in literature reviews for stakeholders other than researchers that participate in the evaluation. To this end, we disclosed *Highlight&Go* to the public at the end of 2017. Specifically, we uploaded *Highlight&Go*, a video, and a user manual to the Chrome Web Store. In addition, a paper about *Highlight&Go* have been published at the EASE'19 conference (International Conference on Evaluation and Assessment in Software Engineering)

[DMA19], and a presentation has been conducted at the IAnnotate conference (Annotation conference organized by Hypothes.is, a foundation that enables annotation in research), and a demo session was provided at JISBD'18 (Spanish Conference on Software Engineering and Information Systems). Once the attendees were back home, we waited to see whether they found *Highlight&Go* useful to the extent of installing it. On December 2021 (see Fig. 3.16), *Highlight&Go* is enjoyed by almost 60 users.

Software installation is regarded as a proxy for utility. It can be argued that the discretionary effort of installing *Highlight&Go* provides evidence of enough perceived utility that it is at least of interest. Furthermore, the fact that the number of users has been increasing for more than three years points to sustained interest as evidence of utility. An example is a research group from the Institute of Linguistics at the University of Utrecht that contacted us asking for support after they found the extension in the Chrome Web Store. After exchanging some emails, they mentioned that one of the reasons they decided to use *Highlight&Go* was the possibility of obtaining a resultant spreadsheet with all the necessary quotes just by highlighting.

3.8.3 Derivation of design principles

So far, we looked at *Highlight&Go* as a whole. Now, we disentangle the distinct mechanisms that are responsible for the utility detailed in Section 3.7. Table 3.6 outlines the main design principles. Design principles reflect knowledge of both IT and human behavior [GKS20]. Accordingly, a design principle should provide cues about the effect on the target audience (i.e., classification of primary studies), the technological cause (i.e., incorporating automation to data extraction), and how it is instantiated in the solution (i.e., incorporating a color-coded highlighter that supports extraction forms and transcripts classification decisions into a spreadsheet).

3.9 Summary of the ADR process

Table 3.7 summarizes the revised set of ADR principles resulting from this research project for problem formulation (principles 1 and 2), organization-dominant BIE (principles 3, 4, and 5), reflection and learning (principle 6) and formalization of learning (principle 7).

3.10 Conclusion

Data extraction is one of the most challenging steps when conducting a literature review. However, it is one of the less investigated, most poorly reported, and most difficult to reuse steps from literature reviews. Different tools are used during the long journey of conducting a secondary study as no tool fits all. For the case of data extraction, spreadsheets are widely used due to their versatility, but translating coding results manually takes a lot of time and most

Table 3.6: *Highlight&Go* as an instantiation of the data extraction Design Model.

Expected Effect on researchers behavior	Technological Cause	*Highlight&Go* instantiation
No transcript needed between PDF and spreadsheet	Automatic data mapping	Transparent extraction of coding events to a spreadsheet
Increase efficiency in metadata extraction	Automatic discovery of metadata in digital libraries	Transparent extraction of metadata
Facilitate observability and traceability of taken decisions	Overview & detail in PS and at SLR level	Navigation over primary study coding & Three level overview for SLR: tabular classification + notes of taken decisions + URL-addressable evidences (i.e., links to annotations)
Reduce incompleteness of data extraction	Extraction status awareness	Font-color semaphore at theme and primary study level
Reduce extraction inconsistency among extractors	Color-coded highlighters	Codebook-based color coding annotation & Consensus awareness using color semaphore

Table 3.7: Mapping *Highlight&Go* project to ADR principles.

ADR Principles	The ADR process in the *Highlight&Go* project	Main Actions
Principle 1: Practice-Inspired Research	Research was driven by the need to provide efficiency and effectiveness data extraction in a Computer Science Research group	Identified based on the literature the practice of data extraction and making exploratory interviews to researchers that have conducted Secondary Studies in our research group.
Principle 2: Theory-Ingrained Artefact	The main theories that inform this research: ontology for SLR portability [dABMN+07, YYH+12, W3C17] and SLR tooling requirements [AZCHH17, GF17]	Revision on Secondary Studies literature
Principle 3: Reciprocal Shaping	Researchers and Web Engineers teamed up to scope and shape the intervention	*Highlight&Go* unfolded through prototyping
Principle 4: Mutually Influential Roles	The ADR team has included two Web Engineers, one evaluation specialist and two practitioners	Sharing meeting to consider technical feasibility of practitioners suggestions
Principle 5: Authentic and Concurrent Evaluation	Real evaluation episode conducted throughout 3 months	Data extraction for a Systematic Mapping Study and a Systematic Literature Review were conducted.
Principle 6: Guided Emergence	The preliminary design of *Highlight&Go* was continuously reshaped thanks from feedback provided by SLR reviewers	Four major insights were provided based on reviewers observations (Section 3.7.1)
Principle 7: Generalized Outcomes	A set of design principles for to increase data extraction portability, effectiveness and efficiency was articulated	*Highlight&Go* was made publicly available through the Chrome Web Store, videos and manuals are made available online.

researchers do not capture traceability data, so the process followed is not repeatable and auditable, not only by researchers themselves but also by third researchers. We advocate the use of W3C Annotation recommendations as a driver of portability, efficiency, and effectiveness. We introduced requirements to improve researchers' productivity in data extraction by integrating color-coded annotation with spreadsheets in *Highlight&Go*. A qualitative and quantitative evaluation has been conducted to evaluate the solution where results are positive to speed-up data extraction, but also to make the literature review's results auditable. Making data extraction traceable thanks to web annotations could lead to researchers on how data extraction is conducted.

Part of this chapter has been published:

- Díaz, O., Medina, H., & Anfurrutia, F. I. (2019). Coding-Data Portability in Systematic Literature Reviews. Proceedings of the Evaluation and Assessment on Software Engineering - EASE '19, 178–187. **CORE A, Class 3**.

And is under review in Information and Software Technology Journal:

- Medina, H., Azpeitia, I., Anfurrutia, F. I., Díaz, O. Supporting efficient and effective traceable data extraction through annotation tooling.

Chapter 4

Web annotation for assignment marking: *Mark&Go*

4.1 Introduction

In the previous chapter, we advocated for the use of web annotations for data extraction in secondary studies (e.g., Systematic Literature Reviews). We have seen web annotation as a purposeful mechanism to increase the efficiency and quality of extracted data where evidence is annotated in a manuscript to provide a classification based on a type of codebook (e.g. data extraction form). This practice is not that far from what is done in assignment marking (also known as assessment in education). In this context, the goal is to find evidence over a student assignment or an exam to decide on a mark based on a set of criteria (e.g. Evaluation Rubric). In the educational area, the European Space of Higher Education brought about an increased focus on continuous assessment.

Continuous assessment adds to grading the concern of tracking student progress and (re)acting accordingly. It "provides feedback to both lecturers and students, allowing the former to make some strategic decisions like changing the type of exercises, while also allowing the latter to regulate their study time" [PLCPYC16]. Thus, the feedback's goal is not limited to grading students but gaining insights about students' healthy progress. Quality feedback is then a cornerstone of continuous evaluation whose value comes from being both *timely* (i.e. provided in time to improve the next assignment), and *idiosyncratic* (i.e. referring to what is already known about the student while pointing to instances in the student's assignment where the feedback applies) [Nic10].

These are nice-to-haves. However, high student numbers and reductions in staff-student ratios frequently mean that quality feedback is just not feasible [Hux07, MT17, Cam05]. This is a significant issue, as inadequate feedback

practices can reduce students' opportunities to achieve learning outcomes, which may decrease both motivation and performance and lead to attrition [MS14, RFK19].

In this context, we look into web annotation tools to provide quality feedback at scale. Hence, the main premise of this chapter is that

> *assignment marking can be more efficient and effective if conducted with the help of annotation tools that account for the assessment practice in a continuous evaluation setting in higher education*

This chapter introduces the practice and the problem of assignment marking in higher education, introducing the specific case of the Computer Science Department of the University of the Basque Country. Then, following the ADR method we introduce the conducted two ADR cycles presenting requirements for a customized web annotation tool and its instantiation in *Mark&Go*, an annotation tool for color-coded highlighting of assignments marking that is built on top of Moodle. The aim is to improve the efficiency of providing effective feedback to students.

We start by profiling the practice and formulating the problem.

4.2 Problem formulation

Problem formulation in ADR is drawn by practice-inspired research and theory-ingrained artifact [SHP+11]. The former reflects the premise that IT artifacts are ensembles shaped by the organizational context. In this case, the organizational context is the lecturers at our university. The latter highlights that ADR does not stop at identifying the problem, but also provides an intervention to alleviate this problem. Before describing the practice of lecturers at our university, we will start by describing the feedback dilemma in Higher Education.

4.2.1 Feedback dilemma

The feedback dilemma gathers two main aspects of feedback, effectiveness and efficiency. We start by describing what is effective feedback.

Effective feedback. Poor feedback is reckoned to cause: (1) a decrease in students' learning motivation [Her12], (2) untapped feedback [Bik14, AG10], and (3), an increase in students' anxiety and shame [DG17]. No wonder quality feedback in education has been comprehensively studied [HWH20, PJD+19, Shu08, HT07]. Specifically, Nicol [Nic10] provides ten recommendations for well-written feedback. We introduce here those that are not related to the language itself (e.g., understandability) but those related to feedback semantics, namely:

- *personal* feedback, i.e., referring to what is already known about the student and their previous work,
- *specific* feedback, i.e., pointing to instances in the student's assignment where the feedback applies,

- *contextualized* feedback, i.e., framed with reference to the learning outcomes and/or assessment criteria.

Efficient feedback. *Timeliness* is a particularly prominent quality aspect participants note across many studies [LC18]. Nicol also includes it as a critical element of quality feedback, specifically to achieve *developmental feedback*, i.e., the extent to which students consider that they can use or apply the feedback provided in the following assignments [BM13, DHM+18]. In their study on students' perceptions of lecturer feedback, Lizzio et al. [LW08] reported that developmental feedback was most strongly associated with students' evaluations of effective assessment feedback, beyond encouraging and fair feedback. As a result of a questionnaire survey conducted with academics and students at Liverpool John Moores University, Mulliner et al. [MT17] indicate that 15 working days (two to three weeks) was found to be acceptable and realistic to the greater majority with regard to mid-module assignments, group work, and laboratory work. Continuous evaluation makes developmental feedback even more critical. In this scenario, assessment aims at not only grading students but gaining insights about students' healthy progress, and providing appropriate feedback for students to outperform in subsequent evaluations [DHM+18].

The dilemma. Nicol [Nic10] recognizes that meeting all factors of quality feedback (i.e., personal, specific, contextualized, and timely) is a challenge to implement in mass higher education, where student numbers are large and regular personal contact is difficult. In the same vein, Boud et al. [BM13] observe that the practical dilemma of higher education is that the amount and type of feedback that can realistically be given is severely limited by resource constraints and, of course, the tradition and expectation of not "spoon-feeding" students. This concept has been referred to as the *Iron Triangle* (also known as the triple constraint) [DKUT09, IJG08, RFK19]. The logic of the Iron Triangle implies that the three triangle vertices of access (i.e., the access of students to higher education), cost (i.e., the cost of assessment in terms of time or personnel), and quality (i.e., the quality of conducted assessment) are locked in an unbreakable relationship, such that making changes to one or two of the vertices will inevitably have an impact on the third. Specifically, the quality of teaching, learning, and assessment is constrained when access to higher education increases. In a setting characterized by high student numbers, instructors might know the theory about quality feedback, but they may not be able to afford it. This is not aided, of course, by the fact that correcting and evaluating students is being reported as prone to procrastination among lecturers [LFF19]. The argument then goes as follows: (1) procrastination is a main stumbling block for timely feedback timely; (2) untimely feedback jeopardizes developmental feedback; and (3), eroding development feedback risks the feedback losing some of its educational potential. This vindicates the need for interventions that act upon procrastination in an assessment-feedback scenario.

The next section outlines how this dilemma arises among lecturers at our University.

4.2.2 Practice-inspired research

ADR starts with a problem perceived in practice. This section examines the practice of *feedback giving in students' assignments* at the Computer Science Department of the University of the Basque Country, hereafter referred to as "the practice".

The practice

The phenomenon under investigation is "student assignment marking". We study this phenomenon in terms of frequency and impact. For the former, we consider the number of students and the number of evaluation episodes. A lecturer who has 30 students and conducts 4 evaluations, experiences this phenomenon 120 times. As for the impact, we consider the correction elapsed time and timeliness, i.e., the number of days between uploading the assignment and reporting the marks.

Fig. 4.1 depicts the practice as a state-transition diagram. It starts with the submission of the assignments by the student. Next, the correction activity involves highlighting, commenting, and grading the student's documents. Finally, the lecturer reports the grade and provides feedback.

To gain situational knowledge, we conducted an exploratory survey[1]. We have sent a call for participation in the Department of Computer Languages and Systems in the three campuses of the University of the Basque Country. Thirty lecturers participate in the exploratory survey (n = 30). The purpose was to instantiate the marking state-diagram for our department (see Fig. 4.1). We request lecturers to quantify the effort they usually spend in each of the activities in the assessment process during a course [2] (see Fig. 4.1). To this end, we asked about the number of submissions during their course, the time invested in correction and reporting and the elapsed time from students' submission to the publication of feedback.

Submission-wise, lecturers reported the number of students[3], where 26.7% has less than 25 students, 20% more than 50, and most of the lecturers have between 26-50 students per course. We asked lecturers about the number of assignments students submit during the 4-months course. Results were distributed in tertiles: 33% of the courses have less than or 5 evaluation milestones, the second tertile is between 6 and 10 assignments, and the third tertile shows that 33% of lecturers have to correct more than 10 assignments per course.

Correction-wise, we asked lecturers to measure the number of assignments that can be corrected in an attention span of 90 minutes[4]. Results show that

[1] Complete survey results available at rebrand.ly/LSIFeedbackQualitySurveyResults

[2] In case a lecturer gives classes in more than one course we asked them to think about the one with the heaviest load when responding to the questionnaire.

[3] We asked about ranges: less than 25, between 25 and 50, and more than 50, as the classrooms at our faculty have a maximum capacity of 25 students

[4] Average person's attention span is around 90 minutes: https://www.csuohio.edu/writing-center/managing-your-work-load-1

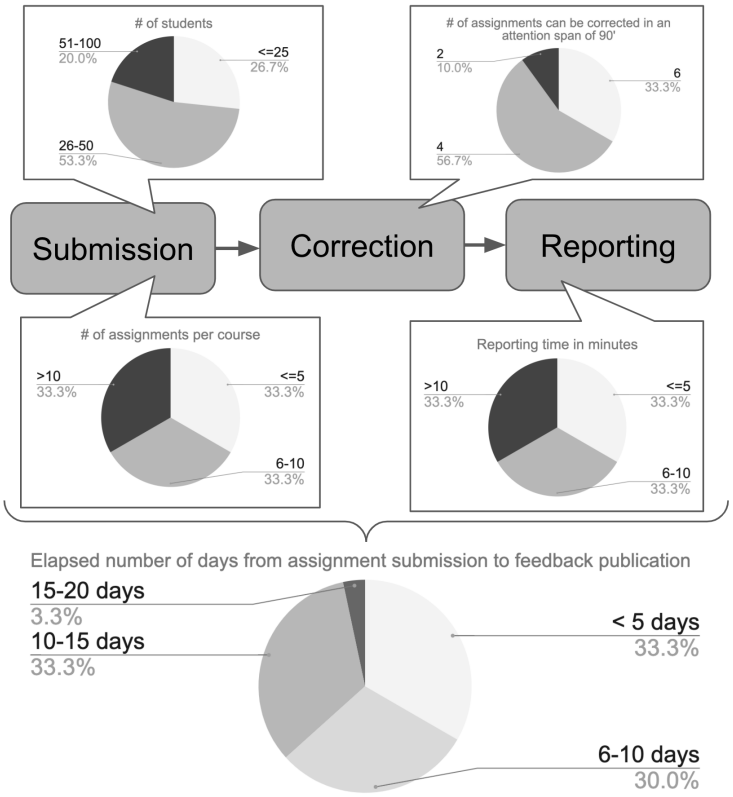


Figure 4.1: Assessment process (submission, correction and reporting), its load at our department and elapsed time to conduct the whole assessment process. Results are for the 30 lecturers surveyed.

33.3% of lecturers are able to correct 6 assignments in 90', while 56.7% correct up to 4 and 10% are able to correct two assignments.

Finally, **reporting**-wise, we captured the time invested per assignment to report the marks in Moodle. We have divided results into tertiles where the first tertile invests 5 or less than 5 minutes, the second third of lecturers spend between 6 and 10 minutes, and 33.3% of lecturers invest more than 10 minutes uploading marks and comments to Moodle.

In addition, we wanted to assess feedback timeliness. Based on scales proposed by Mulliner et al. [MT17], Fig. 4.1 depicts the results: 33.3% provide marks in less than 5 days; 30% between 6 to 10 days; and 3.3% exceeds 15 days.

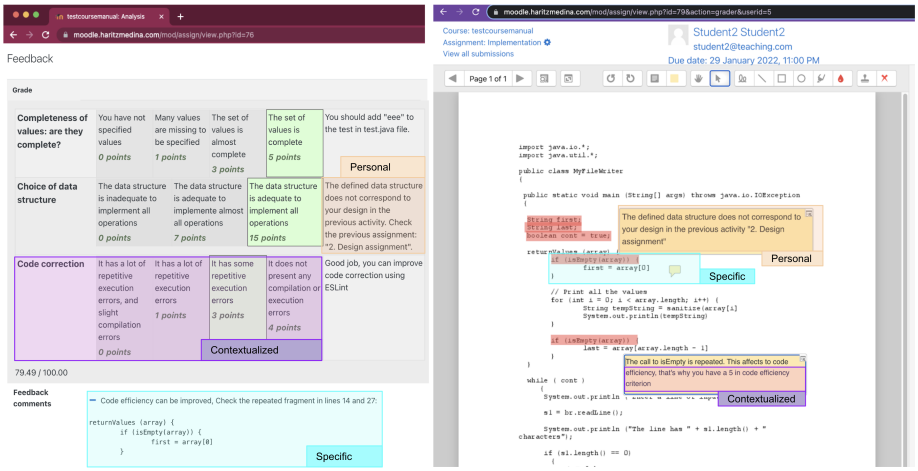


Figure 4.2: Moodle's means for providing feedback based on the student's assignment: rubric-situated (A) vs. assignment-situated (B).

The context

The phenomenon does not take place in a vacuum. Context is paramount to diagnosing the problem and assessing the extent the solution can be generalized to departments other than ours. We consider three key contextual factors that may impact the practice: the sort of assignment, the course where the assignment occurred, and the tooling involved.

The sort of assignment. The very same practice might deliver different results depending on the sort of assignment to be assessed. Specifically, we envision two factors that might have an impact on developmental feedback. First, the existence of a *rubric* might help structure and drive the feedback process. Second, *the sort of assignment* also influences the effort and characteristics of the evaluation. We focus on *textual* assignments. This applies to open questions or code snippets. We do not consider here questionnaires nor assignments that require some sort of drawing (e.g., UML diagrams). Although certainly important, we do not consider here the psychological features of lecturers that can make them procrastination prone. The difficulty and ethical issues that such characterization would have implied advice not to consider them in this first intervention.

The sort of course. Factors of the course that might influence development feedback include *the course's enrollment figures* and *the continuity factor*. First, the larger the number of students, the larger the feedback effort. Second, the larger the extent continuous evaluation is followed, the larger the importance of providing timely feedback for students to benefit for the next assignment. The strength of the continuous evaluation factor is measured in terms of the

perceived need for correctly conducting an assignment depending upon a proper understanding of the concepts assessed in the previous assignment. The larger the dependency, the more important developmental feedback is.

The tooling. We use Moodle for (1) the submission of assignments by the students, (2) the display of evaluation criteria (i.e., the rubric), and (3) the dissemination of results to the students. The latter is achieved through Moodle's feedback page (see Fig. 4.2). For each student, this page positions the student along with the rubric grading by shading the cell that stands for the student's punctuation. In addition, the lecturer might provide feedback. Fig.4.2 shows the case where feedback is framed with reference to the rubric concern (*contextualized*), pointing to paragraphs in the student's assignment where the feedback applies (*specific*), and including a reference to a previous assignment delivered by this student that somehow relates with the assignment at hand (*personal*). It could be concluded that this feedback follows Nicol's guidelines [Nic10]. The current practice is for these feedback pages to be *manually* provided on a per-student basis. No wonder, it is odd to come across such complete feedback samples. The most common practice is for lecturers to shade the rubric cell, leaving all the comments in the margins of the paper-kept assignments [DJtBAvD21]. Unfortunately, most students limit themselves to seeing the results in Moodle. If they pass the assignment, students tend not to delve into the rationale any further. If they fail, a few come over to the lecturer's office to benefit from the feedback. Cultural grounds make Spanish students reluctant to come over to revise their assignments. This experience has also been reported in other settings [Ada14]. The bottom line is that most of the marginal notes laboriously crafted by the lecturers are never read by their addressee: the students.

The problem

The phenomenon under investigation, i.e., "student assignment assessment", seems to be demanding enough even with limited feedback. The promotion of quality feedback should take this reality into account. Any intervention should balance benefits vs. costs. In the search for such trade-offs, this work introduces two premises:

- Moving feedback to the web reduces students' hurdles w.r.t. paper-based office-situated feedback.
- Moodle as a convenient channel for feedback. Rather than providing a separate mechanism, it makes sense to include feedback as part of the Learning Management Systems (LMS).

To substantiate the discussion in concrete examples, two approaches to feedback given via Moodle were presented to the 30 lecturers: rubric-situated (A) vs. assignment-situated (B) (see Fig. 4.2). On the one hand, we questioned the lecturer board about how they observed the implementation of three quality feedback attributes (specific, contextualized, and personal) in two Moodle

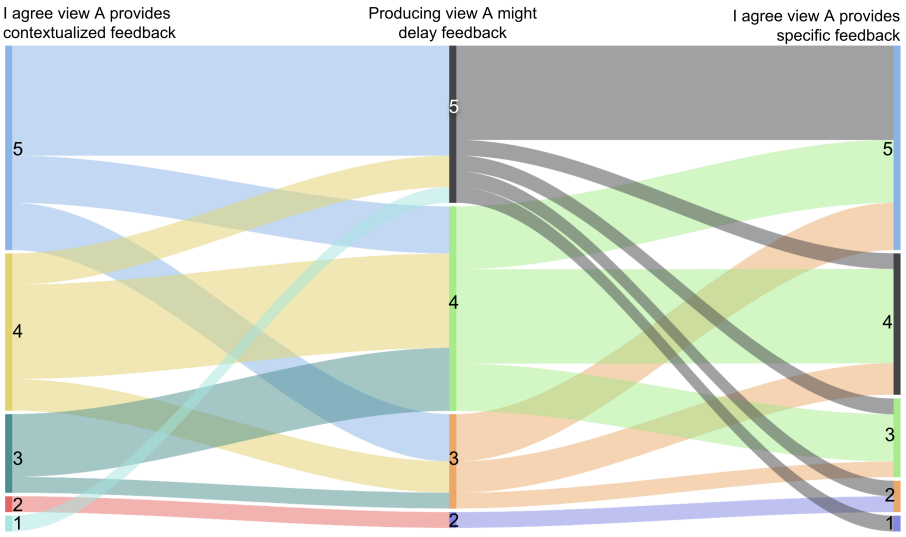


Figure 4.3: Lecturer board opinion in a 5-point LIKERT scale about implementation of contextualized and specific quality attributes in Fig. 4.2 (A) and their correlation with agreement that producing that feedback would delay assessment providing.

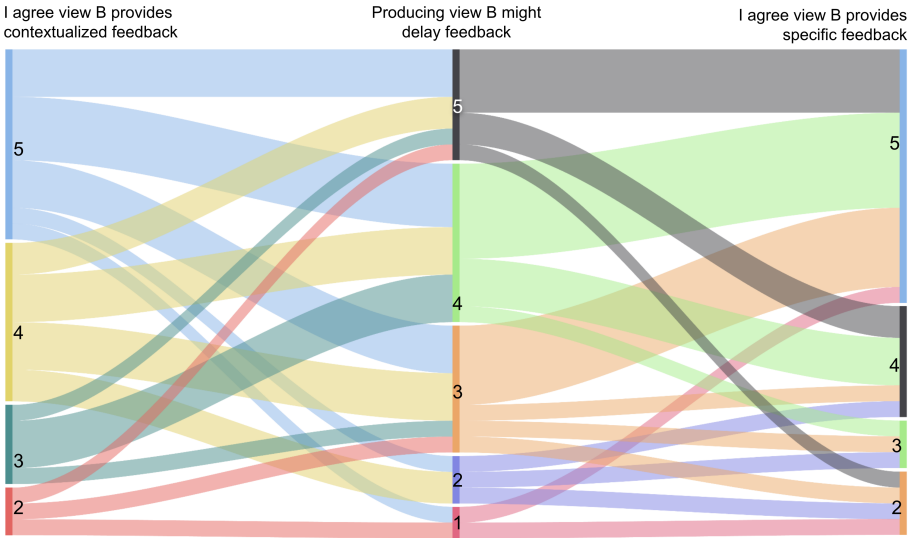


Figure 4.4: Lecturer board opinion in a 5-point LIKERT scale about implementation of contextualized and specific quality attributes in Fig. 4.2 (B) and their correlation with agreement that producing that feedback would delay assessment providing.

examples. There is no clear winner, but both cases agree that examples address contextualized, specific, and personal feedback. On the other hand, we questioned to what extent they have to invest more time to produce the kind of feedback in the examples (i.e., producing that feedback will delay the feedback). Most of the lecturers agree that producing any of both feedback might delay feedback. Figs. 4.3 and 4.4 show the correlation between variables, contextualized and delay and specific feedback and delay.

Fig. 4.3 shows the correlation between the level of agreement of contextualized feedback and that producing the feedback in the view A (i.e., rubric-situated view) will delay the reporting time, and on the other side the correlation between the level of agreement of the view A accounts for specific feedback and that producing the feedback will delay it. The same correlations are shown in Fig. 4.4 but for the feedback in view B (i.e., assignment-situated). We have calculated the Spearman correlation for the four combinations between variables [Spe87]. The Spearman correlation coefficient, r_s, can take values from +1 to -1. A r_s of +1 indicates a perfect association of ranks, a r_s of zero indicates no association between ranks and a r_s of -1 indicates a perfect negative association of ranks. The closer r_s is to zero, the weaker the association between the ranks [Ako18]. For our case, lecturers that agree on the feedback is contextualized in the view A and that it would take more time is r_s=0.704 and p=$8.14*10^{-5}$ meaning that there exists a strong correlation and the result is significant. In the same way, for lecturers that agree the feedback in view A accounts for a specific attribute and it would delay the feedback is r_s=0.608 and p=0.001, meaning a strong correlation too. For the view B, results for correlation between contextualized and delay variables is r_s=0.517 and p=0.02 indicating moderate correlation, and between specific and delay is r_s=0.646, p=0.001, which indicates a strong correlation.

The results indicate that lecturers agree that the way how feedback is provided in both examples addresses contextualized and specific feedback, but it would take more time to produce it, jeopardizing timely feedback. The only exception, with moderate correlation, is for providing specific feedback in the Moodle annotation tool, probably because it only requires highlighting the student assignment. To solve the dilemma between quality feedback and timely feedback, we advocate for the use of tool support to facilitate quality feedback providing.

This leads us to the following research problem:

> How to design a dedicated annotation tool
> that satisfies seamless integration with LMSs
> so that lecturers can increase the feedback quality
> in higher education at scale?

4.2.3 Theory-Ingrained Artifact

This work is based on two main theories. The theory of good feedback described in Section 4.2.1 and Cognitive Behavior Therapy (CBT) to reduce procrastina-

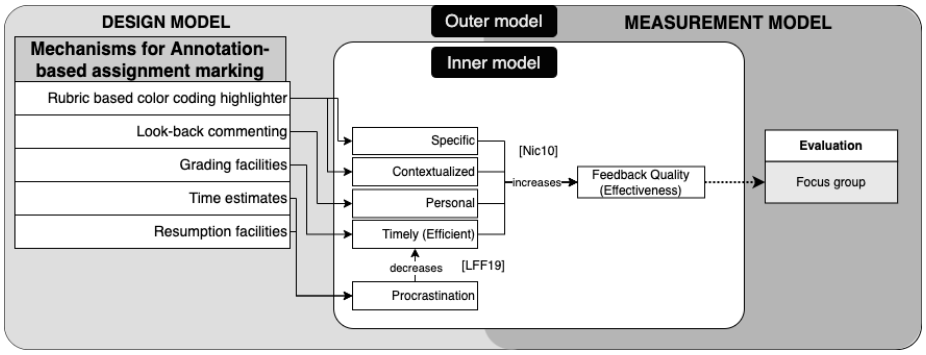


Figure 4.5: Inner-outer model for *Mark&Go*.

tion and provide timely feedback.

To tackle timely feedback, procrastination should be addressed. Procrastination is the avoidance of doing a task that needs to be accomplished by a certain deadline [Ste07]. Procrastination is widely acknowledged as a self-regulation failure. CBT is considered to reduce procrastination more strongly than other types of interventions [vEK18]. Interventions share the same objective (i.e., reducing the intention-action gap). Yet, they choose different paths to reduce this gap [RBF+18]: training self-regulatory skills (e.g., defining time slots or monitoring progress to prevent losing focus on the target task); building self-efficacy (e.g., changing negative and inhibiting thoughts into positive and motivating thoughts); or organizing social support (e.g., peer support).

4.3 Building, Intervention, and Evaluation process

Fig.4.5 introduces the main variables from two justificatory theories: studies on quality feedback [Nic10, RFK19] and the CBT for procrastination prevention [Ste07, LFF19]. The intervention seeks to act upon the independent variables (e.g., contextualized) on the expectation to act upon the dependent variable (i.e., quality feedback). The challenge is how to find a balance between the distinct independent variables along with the dilemma observed in Section 4.2.1. If we put the stress on effective feedback by pushing lecturers to gain in "specific", "contextual" or "personal", then this might result in a detriment for the feedback to be "timely". And the other way around: prompt feedback should be achieved without disregarding the other quality factors.

This theory is tested out through a purposeful artifact developed in an ADR setting in three iterations (see Fig. 4.6) where two lecturers participated in the customization process of the annotation tool for assignment marking while testing it in a real environment (i.e. assessing students assignments in their courses). Finally, the resultant annotation tool was evaluated by 4 lecturers

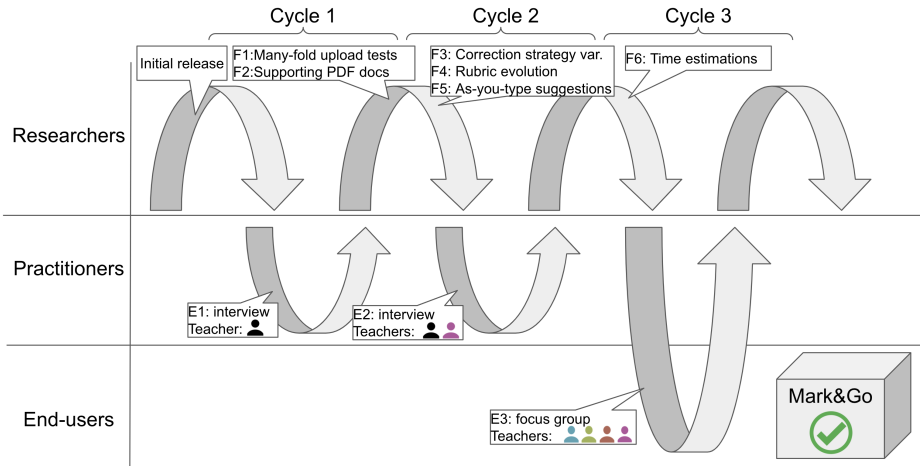


Figure 4.6: Evolution of the *Mark&Go* project. Y axis stands for the members and stakeholders involved in some of the ADR phases. X axis stands for the evolution in time along with the three main cycles.

taken from the department, where they used the tool in a real setting and a confirmatory focus group was conducted to gather qualitative opinions about the use of *Mark&Go* and its impact on assessment.

The next sections delve into the details. We start by describing how the building phase was conducted, to account for feedback quality at scale.

4.4 Building

Building implies the design of an IT artifact based on the problem frame and the theoretical premises adopted. In *Mark&Go* project we act upon four of the quality feedback attributes defined by Nicol [Nic10]: specific, contextualized, personal, and timely. We start by building for specific and contextualized feedback.

4.4.1 Acting upon specific & contextualized feedback

By specific feedback is meant pointing to instances in the student's assignment (i.e., text paragraphs in the assignment) where the feedback applies. In addition, this feedback is contextualized if it references the assessment criteria at play [Nic10]. If this assessment is rubric-based, then these instances are qualified by the rubric item at hand.

Rubric-based assessment can be operationalized through color-coded highlighting. Here, each color stands for a code (e.g., a rubric item), and highlighting becomes the process of mapping text paragraphs in the student assignment to the rubric's items. Color-coded highlighting might account for feedback to be

Figure 4.7: Moodle's rubrics are used to obtain color-coded highlighters. Mouse hover for the grading descriptor to show up. The figure shows the case for the rubric item "Choice of data structure" (yellow code).

specific and *contextualized*. In addition, tight integration mandates for the rubric to be directly obtained from the one held at the LMS. No additional burden should be involved. If you have a rubric, you have a rubric-based highlighter.

Mark&Go generates a dedicated codebook to annotate out of the rubrics kept in Moodle. Once a Moodle rubric page is on display, click on the *Mark&Go* icon in the browser bar. This results in the creation of a highlighter where the rubric's criterion and rubric levels (i.e., the points) are mapped to colors and color grading, respectively (see Fig. 4.7). In the same way, as in *Highlight&Go* themes provides the color and codes' color gradation, themes are mapped in *Mark&Go* as the rubric criterion and codes as the levels pertaining to the rubric criterion. However, where *Highlight&Go* supports classification with more than one code, *Mark&Go* restricts this to a single level (a student cannot be marked with two marks for the same rubric criterion.

From now on, uploading a Moodle-kept student assignment to be graded along this rubric will cause the assignment to be opened in a new browser tab that is decorated with the dedicated highlighter (see Fig. 4.7). Assignments for distinct students can be simultaneously rendered in distinct tabs, where each tab accounts for a separated instance of the *Correction* state. Inside *Correction*, the kick-off state is *Highlighting*.

4.4.2 Acting upon personal feedback

By personal feedback, it is meant to refer to what is already known about the student and their previous work [Nic10]. These references can be included

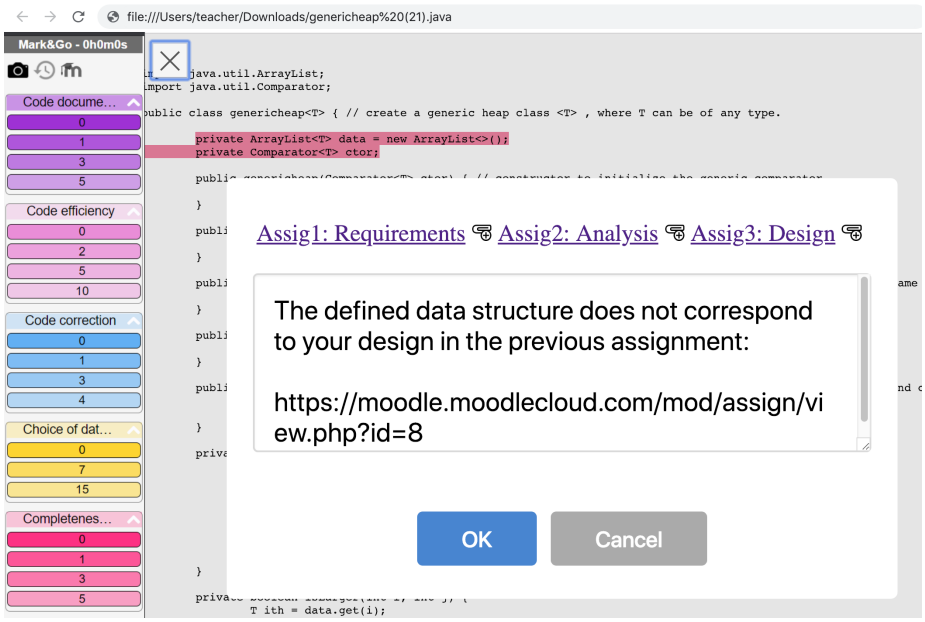


Figure 4.8: *Feedback* state. Feedback involves the interplay of two states: *Highlighting* and *Commenting*. Right click on a highlight for the comment dialog to pop up (a). Look-back commenting is realized by in-placed hyperlink provision to Moodle's previous assignment pages for the student at hand (b). Lecturers can promptly move to these pages to recap on previous comments (c). Also, comments can be enriched with these hyperlinks to make students aware of their healthy progression or repeating mistakes.

along with the lecturer's comments. Personal feedback is akin to continuous evaluation principles where comments should be placed in a broader setting than the assignment at hand. If a continuous evaluation is conducted, it might refer to previous student assignments (look-back commenting). Once again, tight integration mandates for previous student results were kept at the LMS to be readily available at the feedback place. This facilitates lecturers to look up previous assignments in Moodle.

Mark&Go attaches comments to highlights. When opening the comment box double-clicking or right-clicking in any annotated content, *Mark&Go* provides a previous assignment bar (see Fig. 4.8). This bar holds a set of Moodle's URLs to the previous assignments uploaded by the student at hand. This facilitates lecturers to promptly move to the student's previous assignment. Lecturers can include these URLs in their current comments.

The lecturer can go back and forth between the *Highlighting* and the *Commenting* states. The sidebar shows the number of highlights done for each of the criteria. Once enough evidence has been collected for a criterion, lecturers can

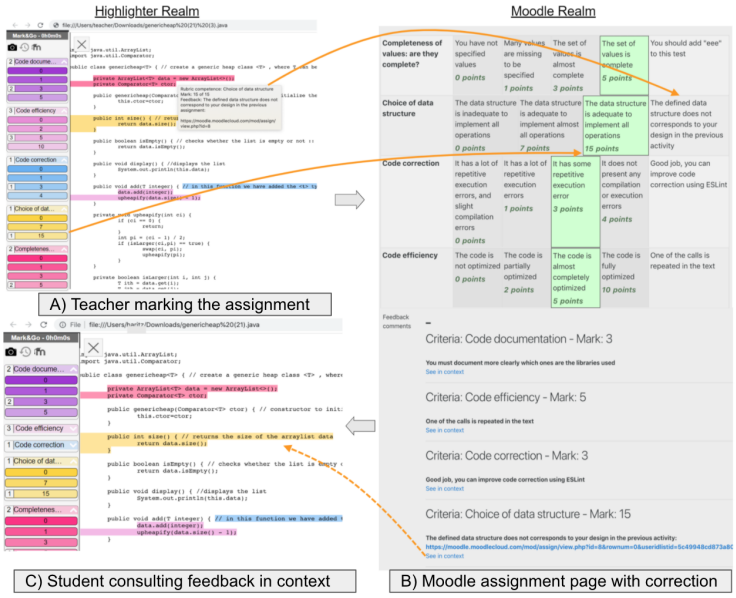


Figure 4.9: *Reporting* state. Feedback activities (a) are automatically backed up in Moodle for student access: Moodle's assignment page with correction results (b), and assignments overlayed with highlights, comments and grades (c).

select the grade in the sidebar. Fig.4.9 (A) shows the case of an assignment with 9 highlights that are distributed along with the 5 distinct rubric criteria. These highlights ground lecturer's grading for each rubric criterion: 3, 5, 3, 15, and 5. It is important to note that highlighting, commenting, and marking normally intertwine. Lecturers can add new comments or revise their grading decisions at any time. Actions on the highlighter are annotations that can be backed up locally (i.e., the browser extension) or remotely (i.e., an annotation server like Hypothes.is). In the same way, marks and comments are updated automatically to Moodle via its API during annotation activity, requiring zero transcription of feedback. We will see in the next section (see Section 4.4.3) that this accounts for timely feedback.

The marking process is transparent to students who access their results as usual through Moodle's feedback page (see Fig. 4.9 (B)). The important point, as commented before, is that this page is gradually and automatically filled up as the highlighting activity unfolds. Fig. 4.9 indicates the mapping between the highlighter realm and the Moodle realm. Students might access assignments there but it is now enriched with color-coded and pop-up comment boxes (see Fig. 4.9 (C)). In essence, this is the lecturer's view but without the update option.

4.4.3 Acting upon timely feedback

We believe that quality feedback will not go mainstream unless there exist some gains for the lecturers. Saving time might be the necessary nudge. Moving the marking activity to the web might not only facilitate feedback but also save time throughout the marking process. Indeed, lecturers especially appreciate the auto-save facility whereby "you mark and go". At the very beginning, we conducted some informal meetings where lecturers other than the ones participating in this project were invited. There was not a fixed agenda. Rather, we wanted to see how lecturers describe their experiences when talking with their colleagues. It was a sort of a surprise that rather than underlining the fancy features *Mark&Go* might have, they stressed aspects, such as mistakes they recurrently make while transcribing marks to Moodle. This became the selling *motto*: *mark & go* and made us realize that for feedback to be provided timely, we should also care for the lecturers' time. We envisage three strategies to pursue timely feedback:

- zero transcription. This is tackled through tight integration with the LMS. Both the input (i.e., rubric, student, assignments) and output (i.e., marks, comments) of feedback is kept in the same place, i.e., the LMS. This is a main non-functional requirement while achieving effective feedback.

- ubiquity. E-feedback is web-based, and this is tackled in *Mark&Go* as assignments, annotations, and marks are on the Internet so never lost or forgotten at home. Ubiquity facilitates marking at odd moments (e.g., commuting). This brings the need for easy assessment resumption.

- procrastination awareness. Correcting and evaluating students' work is prone to procrastination [BP00]. For perfectionists, effective feedback might jeopardize timely feedback. Reducing the feedback burden might alleviate volitional obstacles, yet additional interventions might be needed to combat dilatory behavior.

Next we elaborate interventions introduced to tackle zero transcription, assessment task resumption, and reduce procrastination among lecturers.

Zero transcription using grading facilities. In the same way, as the feedback is created, it must be uploaded to Moodle to report the feedback. This process is manually done, making it error-prone, and looking at the results of the survey in our department 66% of the lecturers invest more than 5 minutes for each assignment only in uploading the results. As *Mark&Go* creates web annotations, those annotations can be used as input to automatically fill up the feedback page in Moodle. Rubric-based web annotation assessment allows automatically matching the selected mark by the lecturer and the rubric level (i.e., the selected mark) in Moodle. Additionally, comments and links to where the feedback applies can be generated using the textual "feedback comment" page in Moodle (see Fig. 4.9). Similarly, we have added a feature that allows you to upload a self-content interactive annotated file (which the lecturer can enable

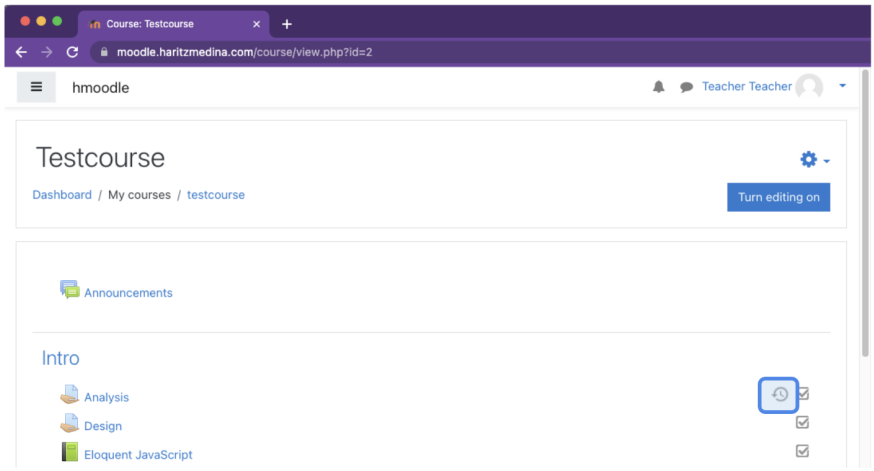


Figure 4.10: Resumption feature. The Moodle's dashboard page is extended with a resumption button. Click on the button for a new browser tab to display the last student assignment.

if necessary). This file includes an HTML-ized version of the student's assignment, the annotations made by the lecturer, and the sidebar with navigation functionality. In that way, the student has access to the annotated document (same vision as the lecturer) without requiring them to install *Mark&Go* to consult their feedback. Without the zero transcription mechanism, lecturers have to spend time uploading marks to Moodle, uploading feedback comments, and (if necessary) the annotated document (e.g., an annotated PDF) manually, increasing the cost of providing quality feedback.

Resumption. Regardless of whether it is conducted on paper or on the web, assessment can rarely be conducted in one shot. This calls for resumption mechanisms that facilitate going back to the last exam/question. Using web Augmentation, *Mark&Go* inlays a resumption button at the Moodle's Assignment page (see Fig. 4.10). In this way, lecturers can move around or switch between tasks, with the certainty that the ubiquitousness of the web and the "resume" button will make going back to correction just a matter of seconds.

Procrastination. CBT regards acting upon the three phases of self-regulation: pre-actional (i.e., missing self-determination concerning the task at hand, and associated with problems in planning and prioritizing tasks), actional (i.e., concentrating on the task and shielding from distractions), and post-actional (i.e., issues arise about low self-efficacy which then, determines the type of self-motivation for the next pre-actional phase) [Zim02, Ste07, KKR08]. We act upon planning by providing an estimate of the time left to finish the correction task.

By moving feedback to the web, correction interactions might be tracked through click analysis. This data might be used to compute distinct estima-

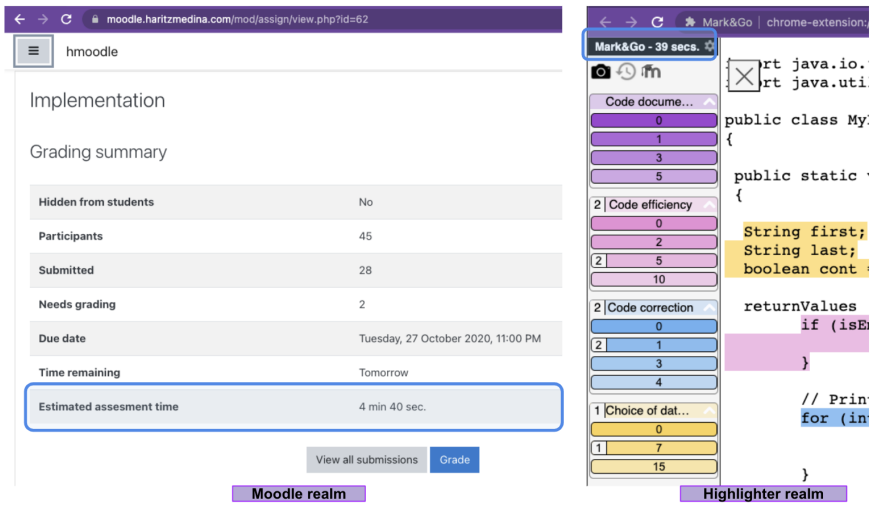


Figure 4.11: Procrastination feature. The Moodle's Grading-Summary page is extended with an estimate about the time left to correct all the students' assignments. The Highlighter realm displays the time estimated to assess the current student assignment.

tions and detect potential procrastination patterns. Using Web Augmentation, *Mark&Go* inlays a time-left estimate onto the Moodle's Grading-Summary page (see Fig. 4.11). This estimate works out based on the time invested in correcting past assignments and extrapolates the time that the lecturer would require to complete the assessment. Estimation accuracy improves as the number of corrected assignments increases. This approach can be extended to more than one course. Since assignments and student distribution tend to be similar between years, past estimations can be applied to the current year, so a fine-tuned estimation can be given about the number of hours it will take to correct the whole set of assignments.

4.5 Intervention

ADR emphasizes continuous evaluation, different from a separate stage of the research process that follows building. While the researcher may guide the initial design, the artifact emerges through the interaction between design and use [SHP+11]. This allows both the researchers as well as the organizational stakeholders to shape the artifact over the research lifecycle.

In the research project, there was an interplay between the development of the theoretical contributions and the development of the e-feedback tool. The theoretical contributions were not only informed by extant theory, but they were also highly influenced by empirical evidence collected from the development and evaluation of the e-feedback tool. In parallel, the design of the e-feedback tool

was based on the emergence of the contributions to theory. Consequently, there was a mutual influence between the emergent theoretical contributions and the development of the e-feedback tool. In total, the case was conducted in three main iterations, where the first two were informed by the practitioners part of the research team. An observational and interview approach has been followed that resulted in 6 main insights that ended up shaping the artifact (see Section 4.6.1).

In the third BIE cycle a confirmatory evaluation has been addressed to measure the real utility of *Mark&Go* as a tool to improve efficiency and effectiveness in assignment marking and examine how the tool has impacted their assessment process (see Section 4.6.2). To this end, we resort to a confirmatory focus group [THB10]. A traditional focus group methodology is used to investigate new ideas where the researcher adopts the role of a "facilitator" and where participants interact among them. A focus group interview is an adequate tool to ask a group of people about their opinion or perceptions about a particular topic, product, or service. The main advantages are the interactive environment where the subjects can discuss each other and are useful to gather depth, strongly held beliefs and perspectives about the tool [CA16]. Tremblay et al. have investigated the adaptation of traditional focus groups in design research [THB10]. They introduced exploratory focus groups to refine the artifact design and confirmatory focus groups to demonstrate the utility of an artifact in the application field (e.g., higher education assessment). The latter one is particularly suitable in our case where the designed marking tool utility should be tested in a real setting. More to the point, focus groups are recommended at the early stage of product development as this is the case of *Mark&Go* project [For08].

4.6 Evaluation

The purpose of the evaluation activity was to analyze the situational knowledge collected from the participating lecturers to develop a proposal for a Moodle-integrated e-feedback utility. The e-feedback tool was iteratively tested to propose changes to the design to shape a better solution. Two participant lecturers that are part of the research team proposed up to six improvements to the tool.

4.6.1 Impact on the tool

This subsection summarizes the main insights provided by the lecturers that ended up shaping the artifact. Frequently, this took the form of mismatches in how the tools map to the lecturers' way of working.

Many-fold Upload Tests. Initially, researchers considered only exams. This premise means that a student only uploads a single file. Yet, the most common scenario in other kinds of assignments is that students could upload more than one file. This turned out to be a 1:n relationship where one rubric might imply the upload of "n" assignment files. In other words, a student's

feedback page might be filled up with the markings of distinct files uploaded by this student in the assignment. It did not arise from the need for an assignment to require "n" rubrics. Therefore, so far, the assessment highlighter has been generated from a single rubric.

Supporting PDF documents. At the very beginning *Mark&Go* was tested in a programming course, where it was only required to provide support for plain-text files (e.g., .java files). However, as it has been introduced in more variety of courses, some lecturers required support for annotation of PDF documents, as nearly half of the assignments were textual written documents in PDF format. One of them also asked for support for other formats, such as Word or Open Documents (ODF), but unfortunately, due to the resource limitations preventing us from tackling this scenario yet.

Correction Strategy Variations. Another concern not apparent at the onset was the existence of distinct correction strategies. Some lecturers preferred to go one student at a time, i.e., fully mark the assignment before moving to the next student. By contrast, other lecturers favor marking one question at a time for all the students before moving to the next question in the assignment. The latter required more effective handling of browser tabs. Lecturers using the one-student-at-a-time strategy were happy with keeping a single tab where the current student assignment was displayed. By contrast, one-question-at-a-time lecturers prefer to have to display over ten assignments concurrently so that they can move from one tab to the next as they mark the same question in all assignments. This required some tuning in both the API calls and tab handling.

Rubric Evolution. We initially considered the rubric to be set in advance. Yet, lecturers expressed the need to adjust the rubric once the marking started. They argued that "some student mistakes might not be fully contemplated at the beginning". They talked about a "rubric calibrating" period. Moodle supports it. That is, Moodle allows the rubric to be changed once the marking has started. Specifically, if a rubric item is changed by introducing-removing new criteria, Moodle shades this item in red and deletes this item's grade for the already-graded students. We need to evolve *Mark&Go* to support this practice. Now, it is possible to (re)generate the highlighter at any time from the Moodle rubric. The grades for the updated rubric items are gone. However, *Mark&Go* does not fully remove the excerpts attached to the previous marks. Rather, it tinges them in gray. Gray is then a reserved color that cannot be used for highlighting but is set apart to indicate previously used excerpts. This facilitates lecturers to spot these excerpts, and if still useful, highlight them again if they are relevant for the new rubric.

As-You-Type Suggestions. Lecturers noted that it was not rare for the comments to repeat for distinct students. Basically, the same mistakes tended to lead to a very similar comment. Hence, lecturers suggested enhancing the comment box with as-you-type suggestions (auto-complete), based on their previous comments. As a result, *Mark&Go* was enhanced to support comment reuse, and a subsequent edition to adapt to the student at hand. This helps lecturers save time and be coherent throughout. However, unexpectedly and most importantly feedback-wise lecturers reported that as-you-type suggestions

motivated them to spend time brushing their comments as it might pay off in subsequent reuses.

4.6.2 Impact on the marking process

The evaluations conducted in the first two cycles facilitated the improvement in the design of the solution. However, it has not been measured by other lecturers in the context of the department to what extent *Mark&Go* is useful to facilitate quality feedback. We conducted a focus group with four lecturers selected from the initial 30 lecturers who participated in the problem definition. Next, we describe the research design and present the main results from the analysis of the focus group.

Research design

Objectives of the study. The purpose of the focus group is

> to discuss *Mark&Go*'s utility for assignment marking, to what extent it has impacted their assessment process and how it has improved feedback quality, limitations of the tool, and future directions or educational scenarios where it can be used

To this end, we have defined the focus group protocol with seven sections in mind from most general to specific questions (see Appendix B).

Participants identification. Participant identification is perhaps the most critical step since the technique is largely based on group dynamics and synergistic relationships among participants to generate data [PT06]. Most of them recommend aiming for homogeneity within each group to capitalize on people's shared experiences. We have selected for the focus group the four participants that have been evaluating the *Mark&Go* tool after the ADR process, which are members of the department. The minimum requirement for selection is that they have worked with *Mark&Go* to assess students' assignments in at least one course in a continuous evaluation and that they have at least two years of teaching experience at the university. Table 4.1 summarizes participants' experience and Table 4.2 the context where *Mark&Go* has been evaluated.

Identify the moderator. Due to the open-ended nature of focus groups, moderation can be complex. For this study, the moderator was one of the authors, while another author played the role of an observer, who recorded supplemental data during the discussion and provided support to the main moderator.

Location. Due to the low availability and distributed location of participants on three different campuses of our university, we resorted to an online focus group [KD13]. An online focus group is not a different type of focus group discussion, but it is applied to the online environment. The session was held using the Webex conferencing application.

Table 4.1: Participants background and experience: number of years as lecturers and current position (full, associate, assistant professor or instructor), category of courses (based on Computing Curricula 2020 knowledge groups [For20]).

Lecturer ID	L1	L2	L3	L4
Years active as a lecturer	20	19	25	2
Current position at the university	Associate	Associate	Full	Assistant
What type of courses have you taught during your career?	Software fundamentals	Software Development	Soft. Fund. Soft. Dev. Systems Modeling	Soft. Fund. Syst. Model.
In what undergraduate programs have you taught during your career?	Engineering & CS	Engineering, Renewal Energies	Engineering & CS	Engineering & CS
Number of assessments per course?	3	6	4	5
Number of courses per year?	3	2	3	3
Active in research projects?	Yes	Yes	Yes	Yes

Table 4.2: *Mark&Go* tool evaluation context: years that they have participated in the evaluation, the number of courses and its category (based on Computing Curricula 2020 knowledge groups [For20]), the number of students assessed, academic course year and bachelor degree (CS for Computer Science and RE for Renewable Energies) and number of evaluation episodes conducted using *Mark&Go*.

Participant	Year	# courses and knowledge area	# students	course year	# eval. episodes
L1	2019 & 2020	2 (Soft. Fund. & Syst. Mod.)	70	1st&3rd (CS)	4
L2	2021	1 (Software Fundamentals)	70	1st (RE)	1
L3	2019 to 2021	2 (Software Development)	20	1st (CS)	4
L4	2020	1 (Systems Modeling)	22	2nd (CS)	1

Table 4.3: Participants' perceptions of the utility of *Mark&Go*'s mechanisms.

Mechanisms	L1	L2	L3	L4	**Total**
Color-coded highlighter	5	5	3	4	**17**
Grading facilities	4	4	4	3	**15**
Look-back commenting	1	3	5	5	**14**
Time estimates	2	2	2	1	**7**
Resumption facility	3	1	1	2	**7**

Data collection

The moderator contacted the four participants after using *Mark&Go* for assessment purposes in a real setting. Permission to conduct the evaluation has been obtained. During the focus group, the researcher acted as a facilitator, posing questions and prompting follow-up questions, encouraging the faculty members to elaborate on certain points and offer additional comments. In Appendix B is presented a short guide followed by the moderator with the main points and questions presented during the session. In addition to the notes taken by the observer, the session's audio was recorded for further analysis.

Analysis

The session lasted 30 minutes more than initially expected, making a total of 2 hours. The session's audio was recorded and transcribed using Sonix[5] and manually revised. A grounded theory approach has been followed and transcriptions were coded independently by two researchers [CS90]. For the analysis, we have used QDAMiner and *Highlight&Go*[6] respectively. We began by creating an initial template with high-order codes and some lower-order codes that focused on aspects of the data quality metrics.

Results and reporting

***Mark&Go*'s utility for providing quality feedback**. One of the main objectives of the focus group was to discuss to what extent *Mark&Go* is useful for assignment marking. Participants were asked to rank *Mark&Go*'s mechanisms from 1 (the least useful) to 5 (the most useful). Table 4.3 shows the results. At the bottom of the ranking are the time estimates and a resumption facility. The little success of the resumption facility might be caused as it is the less used feature, as it is only helpful at the resumption moment, but it is not used further than that. In general, lecturers with more students to assess found it more interesting. However, depending on the assessment strategy followed, its utility is more limited: "I did not find it useful as my correction strategy is to evaluate a criterion in all students before moving to the next, so I usually

[5] https://sonix.ai/

[6] *Highlight&Go* was used to extract text excerpts directly to Google Sheet where a further analysis has been conducted using sheets query facilities.

remember where I left the assessment the last day". In a similar vein, the time estimates were also underscored. However, the reason seems to be that the other three mechanisms are found to be more valuable than time estimation. Usually, the assessment is conducted in the free hours they have during the day (e.g., between two classes, or before a meeting) and they find time estimation useful to decide whether they have enough time to correct the next assignment before "the bell rings and this encourages me to finish the correction on the time that I have". One of the problems they identified in their correction practice is that sometimes they spend too much time because they are too "precise" to provide feedback. They found time estimation also useful to measure themselves: "I use time estimations to measure myself, otherwise I eternalize myself to the assessment".

Looking at the look-back commenting mechanism, there is a bit of disparity in the utility of this mechanism. On the one hand, some participants evaluate it really well as "My current practice is to put lots of comments, so I find it really useful to reuse comments as some mistakes are repeated once and again, and when you start typing a similar comment automatically there appears the comment proposal". On the other hand, some participants indicate that comments are useful, but in their case, it is not that common to make references to previous assignments as they are not dependent or closely related. An unexpected use of this feature is that a lecturer used it to detect plagiarism: "while commenting, I put a comment and I realized that I have commented already another student with the exact rare mistake, so I looked back to check to whom I wrote the comment before and I realized that they copy each other".

On the other side of the coin, participants find the most useful features: the color-coded highlighter, and grading facilities. With respect to the grading facilities, they found the automatic transcription very useful, but not easy to decide on a mark: "Hitting the corresponding mark button and transferring the feedback to Moodle is very very practical, very useful! However, sometimes I found it difficult to decide on the mark as I was highlighting mistakes in students' assignments, and it is difficult for me to spot without looking at the comment included in the highlight of how severe the mistake done by the student is". In the same way, they appreciate the flexibility of combining the different subphases of correction activity, as "you can conduct different correction strategies and combine, highlight, commenting and marking at the same time", "highlighting adapts to what I did before on paper really well". One of the limitations that they addressed is that "I only can highlight the text and I cannot doodle the assignment as much as I would like, sometimes I link different annotations on paper using an arrow and *Mark&Go* doesn't allow me to do that".

***Mark&Go*'s impact in the assessment process**. The second main objective is to know to what extent has changed the impact on the assessment process and assessment result. We asked lecturers about how the tool benefited or not their workflow. Some lecturers mentioned that the tool fits well with their workflow as what they do on paper or digitally is what they can do in *Mark&Go* but with the advantage of comments reuse and automatic translations of the marks to Moodle: "previously I had a document where I write down

the most common mistakes and their corresponding feedback comments to let me reuse them manually, but now *Mark&Go* facilitates me that labor. In the same way, I have my spreadsheet where I write down each mark and later I translate them to Moodle to make them available for the student". However, another lecturer mentioned that it does not fit really well to his practice, as "the assignment consisted of a MATLAB application, I was able to annotate it and mark it correctly with *Mark&Go*, but as it is a software, it requires me to move to MATLAB to execute the program and check that it works correctly".

Looking at the benefits in the assessment process, one of the lecturers who have tried the tool in a course with 20 students saw less benefit to *Mark&Go* as it requires defining a rubric that takes more time, but another lecturer with 70 students replied that "the more students you have the more benefit you can get from *Mark&Go*. You can define a rubric once and reuse it with some adaptations in the next assignments, and as you create a comments pool by just commenting, it becomes better to reuse those comments too. You have to make a small investment at the beginning in configuring the tool and rubric, but later you benefit from *Mark&Go* to speed up the assessment process".

One of the interesting reflections aroused during the focus group is that they are not keen on evaluation rubrics and *Mark&Go* force you to define one, but they found their evaluation rubrics too general "it is a general criterion of what the assignment should have and should not have, however, I feel annotations required you to be very specific pointing exact textual excerpts and it doesn't fit well for all kind of rubrics". One of the lecturers mentioned that "this makes me define a very long rubric at the beginning".

***Mark&Go*'s future directions and improvements**. Lecturers mentioned that possibly, depending on the assessment and course that they will continue using *Mark&Go*. But they mentioned some improvements to support more assessment use cases or improve quality feedback provision using annotations. In some courses, more than one lecturer participates in the assessment, especially in large courses. However, they miss support for multi-lecturer: "in one course that we would like to use *Mark&Go* we could not because we are more than one lecturer assessing at the same time, one to use half of the assignments and the other lecturer the other half. Unfortunately, *Mark&Go* is not ready to be used by more than one lecturer for the same assignment and we couldn't use it".

Lecturers notice a higher satisfaction in students that are really interested in their feedback (which does not always happen), but accessibility to the feedback should be improved: "At the first try only 5 out of 70 students have installed *Mark&Go* by themselves to check the feedback". Based on this fact, we worked on providing for the next course a functionality that uploads automatically to Moodle an HTML file including the student's assignment, annotations made by the lecturer, and the sidebar to let students navigate through annotations, but without requiring to install *Mark&Go*. In this case, the same lecturer mentioned that "the annotated file was used by students because of the improvement in accessibility". Another lecturer mentioned that another improvement to increase feedback accessibility that can be implemented in *Mark&Go* is "to let them

respond to annotations directly with the tool. Students are reluctant to send emails or go to the office to ask for clarification, but maybe letting them answer directly using *Mark&Go* would promote access to their feedback".

4.6.3 Threats to validity

Construct Validity refers to the degree of accuracy with which the variables defined in the study measure the construct of interest. Here, we resort to focus groups to qualitatively evaluate to what extent the implemented mechanisms facilitate lecturers to increase the feedback quality proposed by Nicol [Nic10] (see Fig. 4.5). Focus groups are a valuable instrument to capture group interaction and harness the dynamics involved to prompt deeper discussion and trigger new ideas. But in order for this dynamic to develop it is vital that people's experiences are not well known to each other. The fact that all members belong to the same department might jeopardize this premise.

Internal Validity looks into the extent implemented mechanisms are the ultimate cause of facilitating lecturers to provide quality feedback to their students. From this perspective, other factors besides *Mark&Go* might influence the results. Focus groups can provide rich information as preliminary or larger-scale research. However, they are vulnerable to manipulation by a facilitator and trust between participants as well as between participants and the moderator, to avoid power talkers taking a particular view, where then others are likely to agree even if they disagree [BE11]. To validate the findings of the focus group, we emailed all participants inquiring about their agreement.

External Validity tackles the representativeness of the study and the ability to generalize the conclusions beyond the study itself. First, the number of participants. There were only 4 participants from just one university and all of them assessed using the tool in computer science-related courses, even if the students pertain to different academic programs. The literature generally suggests 6 to 12 participants per focus group. However, [FLT03] posited that participants from a professional background tend to contribute more freely in a smaller group. Reid et al. [RR05] comprised focus groups consisting of just 3 participants with satisfactory data quality. The second is the evaluation context. In this respect, we look for a heterogeneous group where different types of courses, assignments, student group sizes, and different experiences in assessment (from novice lecturers to experienced lecturers with more than 25 years of experience) in the use of the tool (see Table 4.1). Third, the type of assessment conducted and the type of assignments were marked. Most of the assignment types marked are source code (e.g., SQL or XML) or documentation of a few pages where the previous practice, in most of the cases the previous practice was also to annotate, comment, and grade, and this practice maybe is not followed in other assessment contexts.

4.7 Formalization of Learning

ADR intervenes and improves a specific setting through a cycle of making changes by observing the resulting situation and making further changes. We are "experimenting" by making adjustments and observing the effects of those adjustments in assignment marking in continuous evaluation. This improves validity but hinders generalizability and precision. As an ADR project, it should include a reflection of the extent of these threats. In this section, we discuss the generalization of the problem instance (i.e., to what extent the efficiency and effectiveness in assignment marking is a problem for other departments than ours), generalization of the solution instance (i.e., to what extent is *Mark&Go* a solution for the problem) and derivation of design principles (i.e., what sort of knowledge can be distilled from *Mark&Go* project) [SHP+11].

4.7.1 Generalization of the problem

It is worth noticing that the problem at hand is not so much about delivering quality feedback, but quality feedback *at scale*. The labor of providing quality feedback for numerous students within a stringent time frame is the challenge of lecturers, not students. This problem was early noted in the works of Nicol [Nic10] and Boud and Molly's [BM13].

This problem is being reported in 2019 for Australian Universities [RFK19]. When Nicol refers to "the amount and type of feedback that can realistically be given", it can be interpreted as an expression for the need to find a balance between the three aspects that involve the "*Iron Triangle*" [RFK19]: access (number of students that can access higher education), quality (of feedback) and cost (to realize the feedback for the students in terms of time).

Multiple authors have noted that classroom expansion has significant implications for feedback, with increasing workloads and the requirement for quality feedback remaining constant [SM15, Wor18, Car06, HHS01]. This has an impact on students' learning outcomes by affecting the quality of feedback offered or the speed with which it is provided [Gib05]. Specifically, timely feedback was included as a major concern in 24% of the 70 feedback quality studies analyzed by Haughney et al. [HWH20]. In the same way, [Bro17] outlined the timing, amount, content, and mode as the main features for quality feedback. This problem arises for higher education in the majority of knowledge areas, like psychology [LC18] or architecture [MT17].

Some interventions and innovative tools use have been proposed to solve feedback at scale. First, outsourcing assessment. Hiring sessional marking staff [PPMMD17, HP13] or students peer interaction [Tre17]. On the one hand, hiring sessional marking staff requires (economic) resources to be available at the organization (i.e., university). This intervention might result in poorer personalization of feedback as external staff often have limited, if any, opportunities to interact with individual students to sufficiently determine their learning processes and strategies [PPMMD17, RFK19]. On the other hand, student peer interaction only translates the assessment cost to the student [Kun13]. Sec-

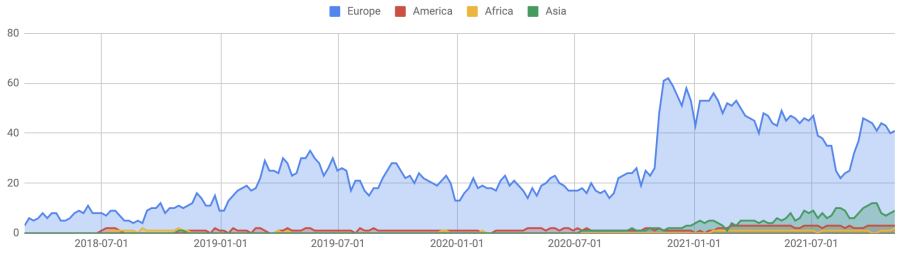


Figure 4.12: *Mark&Go* user base from its release in mid 2018 in Chrome Web Store distributed by region (source: Google Chrome Store).

ond, automated tutor and assessment systems. Several automatic feedback systems have been proposed to reduce the workload on the part of the instructor [CBC+21]. Yet, Cavalcanti et al. addressed that contexts, where automatic feedback can be applied, are limited. The main context for automatic feedback is the comparison with a desired or expected answer: an expected formula in a spreadsheet cell [MJHP12], testing or analyzing source code [SMRA+16, And20, GHJvB17], Automated Writing Evaluation systems [AKKS18] or blueprint feedback [BBW13]. Blueprint feedback maps summative assessment responses (e.g., quizzes, true/false questions, filling the gap,...) to learning outcomes and in that way automatizes feedback reports. Third, learning analytics. Pardo et al. [PJD+19] presented a study in which personalized feedback emails were sent to all students in a large class. This process involved the creation of a discrete set of predefined text-based feedback comments for online activities hosted on a learning management site. The system traces students' behavior in the LMS, interaction with learning materials, videos, and quiz responses. The algorithm compiles the relevant combination of feedback comments based on the students' interaction and sends a personalized email. Fourth, a theoretical study proposes the use of a three-tiered framework for feedback provision: (1) automatic algorithm-led formative feedback, (2) peer-led staged feedback between students, and (3) lecturer-led for assessment feedback [HBKV21].

Previous approaches are promising but imply transferring the control of feedback from lecturers to other people (e.g., marking staff) or algorithms (e.g., automated tutors). In *Mark&Go* project we look for a solution where lecturers retain the control of feedback but are made more productive.

4.7.2 Generalization of the solution

We now address the extent *Mark&Go* might be a *solution* to quality feedback at scale for stakeholders other than our lecturers. To this end, we disclose *Mark&Go* to the public in mid-2018. Specifically, we uploaded *Mark&Go* a video and a user manual to the Chrome Web Store, and provide demo sessions at two conferences: JISBD (Spanish Conference on Software Engineering and

Information Systems) and IAnnotate (Annotation conference organized by *Hypothes.is*, which foundation also enables annotation in education). Once the attendees were back home, we waited to see whether they found *Mark&Go* useful to the extent of installing it. As for October 2021 (see Fig. 4.12), *Mark&Go* is enjoyed by almost 58 users.

Software installation is regarded as a proxy for utility. It can be argued that the discretionary effort of installing *Mark&Go* provides evidence of enough perceived utility that it is at least of interest. Moreover, the fact that the number of users has remained steady for more than two years points to sustained interest as evidence of utility. However, the main threat to the construct validity of this evidence is to interpret installation as real use. It might be the case of users keeping *Mark&Go* installed without really using it. Note also that factors from TAM, UTAUT, UTAUT2, and other models of technology adoption and acceptance, such as perceived usefulness and ease of use, performance expectancy, or effort expectancy, may be significant influencers of the discretionary effort to install *Mark&Go*.

LMS Integration. E-feedback can be framed within the area of e-assessment, i.e., "end-to-end electronic assessment processes, where information and communication technologies are used for the presentation of assessment activity and the recording of responses. This encompasses the end-to-end assessment process from the perspective of learners tutors, learning establishments, awarding bodies and regulators, and the general public" [RGID+09]. From a tool perspective, e-assessment occurs within LMSs. LMSs are all-encompassing software applications for the administration, documentation, tracking, reporting, automation, and delivery of educational courses [ASS18]. Although LMSs aid in the recording of feedback remarks, the feedback process is still carried out elsewhere.

Nowadays, 99% of higher education institutions have an LMS, where 85% of faculty and 83% of students use an LMS [RRG+17]. Moodle is in the top 3 most used LMSs worldwide, with more than 298 million users[7], which accounts the 25% of LMS market share[8]. Our solution instance can, potentially, be used by those users. Similarly, other e-feedback tools account for LMS integration. The cases of *Kami*, *Gradescope* and *Hypothes.is* can be highlighted, as both support integration with more than one LMS. Kami supports integration with Google Classroom, Canvas, and Schoology; Gradescope with edX and Canvas; while Hypothesis supports LTI [IMS10], an approach for LMS interoperability, which allows annotation of PDFs uploaded to almost any LMS that supports LTI.

4.7.3 Derivation of design principles

So far, we look at *Mark&Go* as a whole. Now, we disentangle the distinct mechanisms that on balance are responsible for the usefulness and perceived ease of use as detailed in Section 4.6. Table 4.5 outlines the main design principles. Design principles reflect knowledge of both IT and human behavior [GKS20].

[7] Moodle usage statistics: https://moodle.net/stats/
[8] https://research.com/education/lms-statistics

Table 4.4: *Mark&Go* as an instantiation of the e-feedback Design Model.

Expected effect on lecturers behaviour	Technological Cause	*Mark&Go* instantiation
specific and contextualized feedback	color-coded highlighters	rubric-based highlighter
personal feedback	look-back commenting	ready access to the student's past assignments
no transcript needed	grading facilities incorporated	auto-save on Moodle
better planning	time estimates	estimation about "how much time is left to finish the correction of a student assignment or the whole assignment?"
effortless correction resumption	resumption facilities	back-to-last-assignment button

Accordingly, a design principle should provide cues about the effect on the target audience (i.e., providing personal feedback), the technological cause (i.e., incorporating look-back commenting to feedback), and how it is instantiated in the solution (i.e., a mechanism for ready access to the student's past assignments in Moodle).

Table 4.4 outlines our proposal for e-feedback and its instantiation in *Mark&Go*.

Table 4.5 summarizes the revised set of ADR principles resulting from this research project for problem formulation (principles 1 and 2), organization-dominant BIE (principles 3, 4, and 5), reflection and learning (principle 6) and formalization of learning (principle 7).

4.8 Conclusion

Continuous evaluation makes an assessment to be under pressure in higher education. Lecturers have to juggle to get feedback providing on time (i.e., be available for the student before the next assignment), with enough quality (i.e., to be useful for the student to learn from it and apply lessons learned in the following milestones during the course) to hundreds of students. To improve lecturers' productivity, we advocate moving a step further in e-feedback by applying web annotations. There, LMSs still act as a channel of feedback communication, but it is enhanced with facilities to provide better feedback in less time. We introduce requirements for good feedback and facilitate the plan of lecturers' time to be more efficient and effective. These requirements are tested out through *Mark&Go*, a dedicated annotation tool integrated with Moodle LMS to highlight, comment, and mark students' digital assignments. A focus

Table 4.5: Mapping *Mark&Go* project to ADR principles.

ADR Principles	The ADR process in the *Mark&Go* project	Main Actions
Principle 1: Practice-Inspired Research	Research was driven by the need to provide quality feedback at scale in a Computing Science Degree	Conducted survey about the extent of the problem at the department (n: 30)
Principle 2: Theory-Ingrained Artifact	Two main social-science theories inform this research: quality feedback [Nic10] and CBT [RBF+18]	Revision on student-feedback literature
Principle 3: Reciprocal Shaping	Lecturers & Web Engineers teamed up to scope and shape the intervention	*Mark&Go* unfolded through prototyping
Principle 4: Mutually Influential Roles	The ADR team included two Web Engineers, one evaluation specialists and four lecturers	Sharing meeting to consider technical feasibility of lecturers' enhancement suggestions
Principle 5: Authentic and Concurrent Evaluation	Real evaluation episode conducted throughout two years	A confirmatory focus group has been addressed with four lecturers after been using *Mark&Go* in a real setting
Principle 6: Guided Emergence	The preliminary design of *Mark&Go* was continuously reshaped through use and feedback from lecturers.	Six major insights were provided based on the lecturers' observations (Section 4.6.1)
Principle 7: Generalized Outcomes	A set of design principles for e-feedback was articulated	*Mark&Go* was made publicly available through the Chrome's Web Store. Two videos uploaded

group was conducted to capture the qualitative opinions of 4 lecturers that have used the tool in their courses in the last 3 years. The next follow-on is to evaluate *Mark&Go* from the perspective of the student. Thanks to the use of web annotations, students are able to consume quality feedback online, anywhere at any time, but can also interact with the feedback by replying to lecturers' comments or asking for clarification.

Part of this chapter has been submitted to Elsevier Computers&Education Journal:

- Diaz O., Medina H., Azanza M. (submitted for review in 2022). Balancing Quality and Timeliness in Student Feedback at Scale: A Case of Action Design Research

Chapter 5

Web annotation for peer review: *Review&Go*

5.1 Introduction

In the previous chapter, we advocated for the use of web annotations for the assessment of students' assignments. We have seen web annotation as a purposeful mechanism for assessment. In the same vein, assessment is conducted in other areas such as research. Assessment in research is sustained by peer review, i.e., "the process by which research output is subjected to scrutiny and critical assessment by individuals who are experts in those areas" [Ham12]. It is widely supported by researchers [PRC16], but some opponents claim current reviewing processes to be "slow, costly, ineffective, biased, easily abusable, anti-innovatory or largely a lottery" [Smi10].

Three stakeholders are impacted: authors, reviewers, and journals. Authors are deprived of getting useful feedback to improve their research [Gra, LRF11], often leading to further submissions without modifying manuscripts, and as a result, to a waste of reviewers' effort [Ham12, War11]. Readers consume substandard papers, and, in the worst cases, fraudulent or incorrect work is published due to gatekeeping errors [TDG+17, Aca20]. Finally, journals have their *raison d'être* undermined, i.e., the prompt dissemination and recognition of knowledge advances [Aca20]. Different causes can be blamed for this situation:

- lack of transparency in the process [Aca20, TDG+17]: impossible to trace discussions or reviewers' decisions
- lack of consensus in quality feedback [Ham12, Smi10, TDG+17, War11]: each conference, journal, and reviewer has its criteria to evaluate
- lack of skills and experience [Ham12, LRF11]: especially in novice researchers that require training

- lack of time [Cla10]: it is a highly demanding and time-consuming activity, requiring about 8.5 hours per review

This begs the question of what quality feedback is, and even more, how current annotation tools (e.g., *Acrobat Reader*) can be enhanced with facilities towards quality feedback. Therefore, the main premise of this chapter is that

> *peer review can be more efficient and effective if conducted with the help of annotation tools that account for review specifics*

This chapter introduces the practice and the problem in a peer-reviewing context. Then, following the ADR method we introduce the conducted two ADR cycles presenting meta-requirements for a customized annotation tool, to later instantiate those meta-requirements in *Review&Go*, an annotation tool to highlight manuscripts that guides reviewing and generates a review draft. The aim is to improve the effectiveness and efficiency of the peer-reviewing activity. We start by profiling the practice and formulating the problem.

5.2 Problem formulation

Problem formulation in ADR is drawn by practice-inspired research and theory-ingrained artifact [SHP+11]. The former reflects the premise that IT artifacts are ensembles shaped by the organizational context. In this case, the organizational context is the researchers at our university. The latter highlights that ADR does not stop at identifying the problem, but also provides an intervention to alleviate this problem. We will start by describing the peer review practice.

5.2.1 Practice-inspired research

Peer review is an evolving and heterogeneous practice, with varying approaches depending on its timing and transparency (refer to [TDG+17] for an overview). Broadly, it plays three main roles:

- Quality standard, that is, ensuring the trustworthiness of published research. Peer review helps to distinguish peer-reviewed from non-peer-reviewed literature by providing a kind of "seal of approval" [War11].
- Gatekeeping, i.e., filtering out research that does not meet certain quality thresholds. In this sense, peer review has been described as "the process that routes better articles to better and/or most appropriate journals" [War11].
- Improving work. There is ample consensus among authors that reviewers' feedback helps improve manuscripts [WM15].

Hence, peer review is considered at the heart of scientific communication [Ham12, LRF11]. Yet, it is far from being properly recognized. Reviewers are generally volunteers who receive neither remuneration nor professional credit

[Ham12]. In this context, why do they agree to review? The rationale is manifold:

- Peer review relies on a give-and-take relationship. Most reviewers are also authors that benefit from feedback, and thus try to reciprocate others' reviewing effort [Ham12].
- Peer review is regarded as a responsibility of being part of the community [Ham12].
- Some reviewers enjoy helping to improve papers and seeing research ahead of publication [WM15].

After accepting an invitation to review a manuscript, reviewers are expected to carry out two main activities:

- One or more critical readings of a manuscript, often accompanied by note-taking (i.e., annotation). Before the digital age, annotation was conducted manually, and usually, individually. Now, different tools permit annotating digital content where PDF readers (e.g., *Acrobat Reader*) are one of the most used ones.
- Writing a report. This report should contain (1) an assessment of the strengths and weaknesses of the manuscript, (2) feedback to the authors about ways to improve it, and (3) confidential comments to editors.

Unfortunately, current annotation tools (e.g., *Acrobat Reader*) are general purpose and do not capture the specifics of annotating for review. This leads us to the following research problem:

> How to design a dedicated annotation tool
>
> that provides guidance
>
> so that reviewers can increase the feedback quality
>
> in scholarly peer review?

5.2.2 Theory-ingrained artifact

To design a review-dedicated annotation tool, first, we should characterize peer review in the following terms:

- activities involved. Reviewing process intermingles three key activities: strategic reading, feedback giving, and summarizing,
- actor profile. Reviewing is knowledge-intensive, hence conducted by well-educated people with tight agendas,
- setting. Reviewers continue to be volunteers working under stringent time constraints, but now working online.

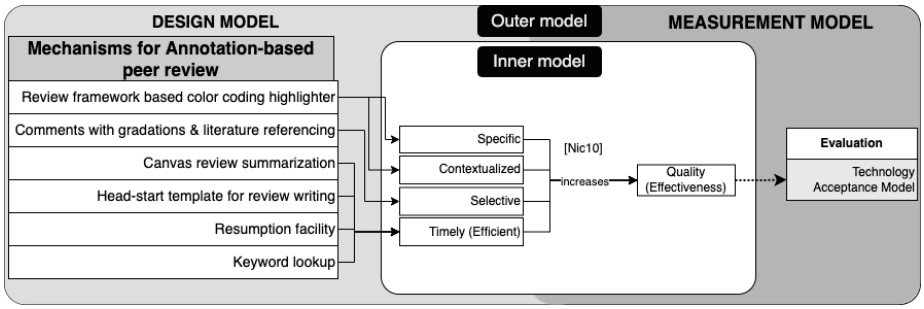


Figure 5.1: Inner-outer model for *Review&Go*.

On reporting about the state of affairs in peer review, Ware noticed that the main areas of discontent include: "concerns at the length of time taken by the process; some concerns at the burdens imposed by reviewing commitments; and concerns about bias and lack of fairness" [WM15]. This reveals a tension between time and quality. Then, the solution should attempt to facilitate timeliness (i.e., making the reviewing process more efficient) without overlooking the feedback quality (i.e., keeping or increasing reviewing process effectiveness).

To provide an answer to these questions, we are informed by theories on providing good feedback [Nic10]. Though initially proposed for student assessment, their principles can also be useful in reviewing settings. Next, we rephrase these attributes for the practice of reviewing (see Fig. 5.1):

- Specific: pointing to paragraphs in the manuscript where the feedback applies
- Timely: provided in time along with the conference/journal deadline
- Contextualized: framed regarding methodological criteria of ample support within the community
- Selective: commenting in reasonable detail on two or three aspects that the author can do to improve the manuscript, distinguishing major concerns (i.e., those that threaten the validity of the study) from minor concerns that can be corrected (e.g., an additional analysis) and providing references that could enhance the work

Fig. 5.2 depicts a conceptual model for reviewing as a process of spotting text paragraphs in manuscripts that later we will instantiate using annotations:

- along with the "specific" mandate, the model places "paragraph" in the middle.
- along with the "contextualized" mandate, paragraph highlighting responds to a purpose: pinpointing evidence of quality. Review quality measurement is about quantifying to what extent a manuscript possesses desirable

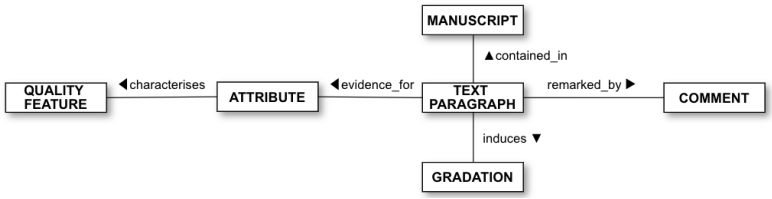


Figure 5.2: Concepts involved in quality feedback.

characteristics. Similar to software quality frameworks, we can distinguish between quality characteristics (e.g., relevance) and their measurable attributes (e.g., adoption and use of the new artifact by real organizations).

- along with the "selective" mandate, highlighting should be supplemented by comments as well as graduation that sets the mood of the comment (e.g., minor vs. major).

In addition, configurability and familiarity become the main non-functional requirements. The former is due to reviewing being a diverse practice [TDG+17]. Even within the same field, the criteria might vary. As for familiarity, smooth adoption advocates for not being disruptive regarding traditional annotation tools. In this regard, our base comparison is with *Acrobat Reader* as researchers in our university are familiar with it.

Next, we identify the following meta-requirements to support efficient and effective peer-reviewing that later will be instantiated in the BIE process along with the concepts involved in quality feedback (see Fig. 5.2).

MR1: Support for specific and contextualized reviews: Review frameworks

Reviewing involves strategic reading, where pieces of evidence in the manuscript (i.e., *paragraphs*) are pinpointed to place them into a review framework to weigh the merits of the manuscript (i.e., *attributes*). This implies the existence of a review framework to measure the quality of the manuscript.

Even within the same field, quality criteria might vary [TDG+17]. DSR is a case in point. In his survey about the quality of DSR, Venable observed that there exists a lack of consensus concerning how research should be assessed [Ven15]. Thus, review frameworks tend to be rather subjective. Although general principles apply, personal preferences and background might certainly tinge on the review. This calls for review frameworks to be customizable.

MR2: Support for selective reviews: comments with graduations and literature referencing

Peer review plays a manuscript-improvement function: providing comments that make the published paper better than the submitted manuscript. About 90%

of researchers overall thought the main area of effectiveness for peer review would be in improving the published paper by providing constructive feedback to the authors [War11, Ham12]. Along with the "selective" mandate, highlighting should be supplemented by comments as well as a "graduation" that sets the mood of the comment (e.g., minor vs. major). Support should be given to capturing these elements as indicators of quality feedback.

MR3: Support for timely reviews: review summarization

Peer review is about grading (i.e., weighing whether merits on balance deserve publication). When it comes to reviewing, the question is not whether the manuscript ticks off all items of the review framework, but whether the manuscript holds enough merits to be worthy of publishing. And merits might not weigh all the same. For instance, good English is certainly a desirable feature. Yet, most authors agree that minor spelling and grammatical errors, though they can be distracting, should not decide the manuscript's fate [Lar11]. More complex is the scenario where the weight of merits depends on the type of work. DSR is an example. Gregor and Hevner illustrate this situation for manuscripts in the invention quadrant [GH13c]. Here, "reviewers find it difficult to cope with the newness". Here, concerns about the design being insufficiently grounded in kernel theories, the design not being rigorously evaluated, or there being no new contribution to the theory made via the design can be excused due to its newness [GH13c]. In a similar scenario, Venable [Ven15] states "a potential resolution that I suggest here is to use a cumulative model that adds up the value of the DSR work's contribution to some (but not necessarily all) of the various criteria, rather than the subtractive model inherent in a check-list approach (where all criteria not met fully count against the research)". This suggests reviewing not merely to gather the manuscript's merits, but a *subjective assessment* of whether existing merits are sufficient. Thus, mechanisms are needed to support review summarization.

MR4: Support for timely reviews: head-start template

As mentioned before, peer review is time-consuming [Lar11]. The overall average (median) time spent by reviewers per article is about 5 hours (mean 8.5 hours) [War11]. Certainly, this very much depends on the manuscript's size, and how detailed the report is. If the report has to be specific, timely, contextualized, and selective, then five hours do not seem that long.

Following the current practice for reviewers where first they annotate the manuscript, take some notes, and once the manuscript is read, they produce the report. This might require reviewers moving back-and-forth between the manuscript and the note editor, threatening the reading focus. In this scenario, a head-start might be provided by obtaining a draft out of the annotations already taken in the manuscript. In a limited manner, *Acrobat Reader* already accounts for this. It generates a text document with a list of the comments upon the PDF at hand [Ado22]. This is a start, although certain limitations

apply: no reference to the comment's target (i.e., the manuscript's paragraphs); no reference to the comment's purpose (i.e., the review frame); no reference to the comment's graduation (i.e., minor vs. major); no a sensible way of clustering comments, just the comments ordered chronologically. This is not a complaint. *Acrobat Reader* is a general-purpose annotation tool, not a dedicated visor for reviewing. However, dedicated visors can go a long way in automating transcript tasks by automatically framing reviewers' comments in terms of the review frame or the graduations.

MR5: Support for timely reviews: resumption facility

Peer review tends to be a fragmented activity. Reviewers do not always find it easy to dispose of 5 hours straight to conduct the review. Support should then be given to resume the reviewing activity. In this respect, ubiquity and offline support are also important since it is not rare for reviewers to work at home, and even when traveling. Facilities should be provided for reviewers to resume the reviewing state before the interruption.

MR6: Account for familiarity

People like to stay in their comfort zone and they tend to be reluctant to fiddle around with new Graphical User Interfaces (GUI) [Pay22]. More to the point, if the tool is sporadically accessed, then users might forget the GUI's gestures from their last interaction. This might well be the case with peer review. According to Ware's survey, active reviewers report an average of 14 reviews per year. This is a bit above once a month, a frequency not high enough to risk convoluted GUIs where reviewers might forget the tool's springs. Chances are reviewers are familiarized with annotation tools (e.g., *Acrobat Reader*). Hence, easy adoption advises custom web annotation tools to mimic *Acrobat Reader* gestures.

5.3 Building, Intervention, and Evaluation process

This theory is tested out through a purposeful artifact developed in an ADR setting in two iterations (see Fig. 5.3, *Review&Go*, where researchers, which are also reviewers, participate in the customization process of the annotation tool for peer-reviewing.

The next sections delve into the details. We start by describing how the building intervention and evaluation of the first cycle was conducted.

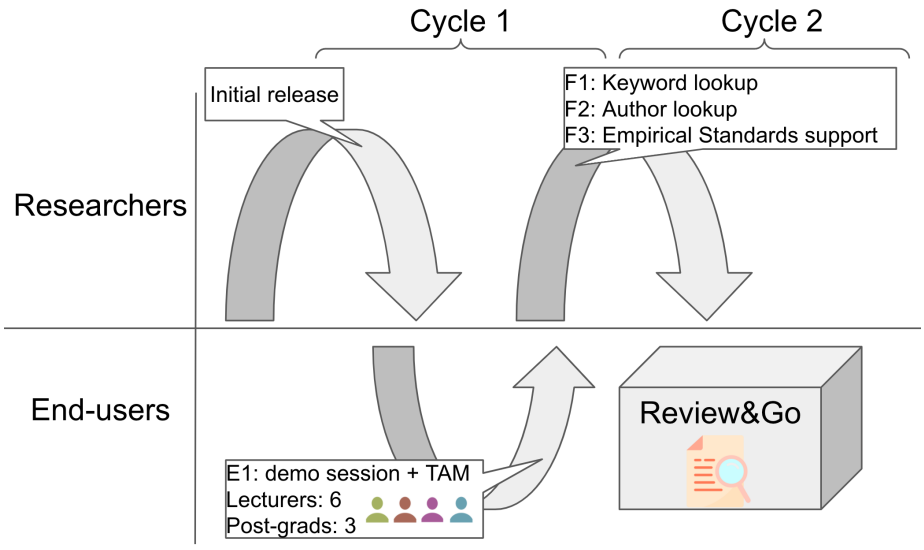


Figure 5.3: Evolution of the *Review&Go* project. Y axis stands for the members and stakeholders involved in some of the ADR phases. X axis stands for the evolution in time along with the two main cycles.

5.4 BIE1: Acting upon performant peer review

5.4.1 Build

Building implies the design of an IT artifact based on the problem frame and the theoretical premises adopted. In the first iteration of the *Review&Go* project, we acted upon specific, contextualized, selective, and timely reviews while keeping the familiarity of the designed software artifact. We start by building the artifact for specific and contextualized reviews.

Support for specific and contextualized reviews: Review frameworks

Annotating for peer-reviewing has a first endeavor: spotting manuscript merits. For this annotation to be strategic, reviewers use their prior knowledge along with clues from the text to construct meaning, and place the new knowledge within a domain-specific frame (e.g., a codebook). These merits are judged along with a quality framework that very much depends on the publication venue. We do not claim this list to be exhaustive, not even correct. *Review&Go* first release offers aspects taken from the DSR community [Ven15] (rigor, relevance, and design) as a first option that can later be tuned to the personal taste of the reviewer at hand. The key point is that *Review&Go* resorts to these quality criteria for color-coded highlighting (see Fig. 5.4). That is, highlighting not only collects but also typifies evidence along with the review framework that is realized as a codebook to let the reviewer classify annotations. This default

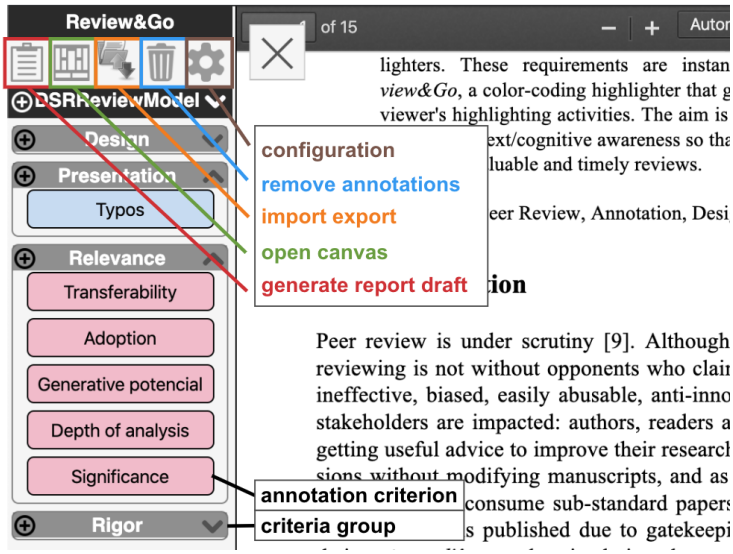


Figure 5.4: Review framework is realized through a color-coded highlighter. Import/export codebook and annotation, canvas view and report draft generator and other utilities are provided in the top-left toolset.

framework can be customized at your wish by changing the sidebar's button labels. As a bonus, a "typo" button permits to spot misprints that will next be automatically listed at the end of the report (see Section 5.4.1). In the same way, we have implemented the functionality to import and export the created review frameworks to support review framework sharing among different reviewers in the same venue.

Support for selective reviews: comments with graduations and literature referencing

Along the conceptual model in Fig. 5.2, quality feedback qualifies evidence (i.e., highlights) through comments and graduations. *Review&Go* permits attaching this information by double-clicking upon the highlight at hand (see Fig. 5.5). Guidelines also recommend complimenting comments with references to the literature [War11]. To this end, *Review&Go* includes a reference finder where typed keywords are passed to the DBLP API3[1]. The reviewer just needs to click on for the full reference to be included in the report (see Section 5.4.1).

Support for timely reviews: canvas review summarization

As the revision progresses, reviewers might need to have an overview of the situation so far. This comes in handy when the final decision should be taken,

[1] https://dblp.org/faq/How+to+use+the+dblp+search+API.html

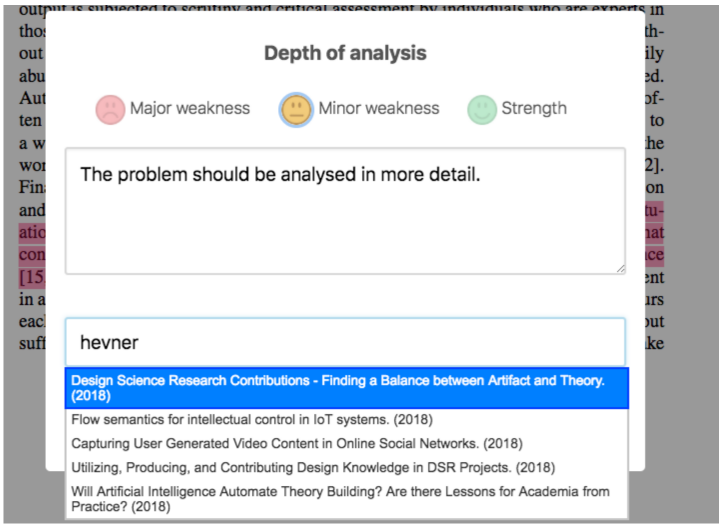


Figure 5.5: Double-click on a highlight for the comment box to pop up. Besides the comment, grades and references can be introduced.

but also if the reading is resumed after some days off. Canvas view is a successful approach for overviewing work in progress in a compact way. Just to mention, it is suggested in DSR projects [JP14] to overview the artifact, the problem addressed, the knowledge base used, etc. By contrast, reviewing is not about constructing but the other way around: "de-constructing" the manuscript, i.e., brushing away the narrative to get the bare essentials of DSR milestones. On these premises, *Review&Go* generates a canvas out of the highlights gathered so far (see Fig 5.6). A glance serves to apprehend which DSR aspects have been tackled, and those that have been left out. Worth noticing that paragraphs have been turned into hyperlinks. On clicking, reviewers can move back to the PDF to see the paragraph in context.

Support for timely reviews: head-start template

Once annotated, the manuscript itself is a good conduit for the review. Yet, using manuscript structure to organize reviewers' comments might not be the most effective way. *Review&Go* supplements PDFs with a report draft (see Appendix D). Two ways to arrange comments are available: attribute-based or grade-based. The former organizes comments along with quality attributes. By contrast, the grade-based option arranges comments by strengths and weaknesses. No matter the way, comments are always framed by the associated highlighted paragraph and the manuscript page which holds that paragraph. Finally, typos and references are added at the end of the draft. The draft is structured following the guidelines defined by Hames [Ham12]. Being text,

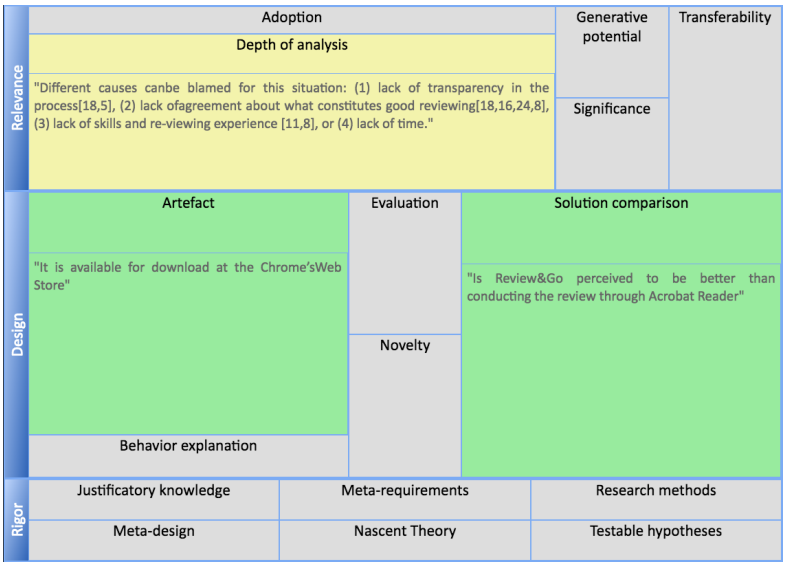


Figure 5.6: A canvas generated out of the highlights: regions stand for quality attributes; content corresponds to manuscript paragraphs; background colors denote the graduation.

drafts can now be copy&pasted into the editor at wish for completion.

Support for timely reviews: resumption facility

A review can be conducted on different days, in distinct places, and even multiple copies of the manuscript. No matter the setting, reviewers should be able to go back to their review at the point they left it. *Review&Go* exhibits some features that facilitate resumption. Being web annotation, *Review&Go* naturally supports ubiquity[2]. Following the W3C Web Annotation recommendation, *Review&Go* faces annotation portability so that annotations can be overlaid on top of manuscript copies other than the ones on which annotations were first conducted. These aspects facilitate going back to annotations (i.e., "the annotation state") but they do not restore "the mental state". To this end, the aforementioned canvas can help. *Review&Go*'s canvas can be obtained at any time to display the annotations collected so far. After some hours/days off, reviewers can display the canvas to restore "the mental state" before the interruption. In addition, a button is provided to navigate to the last annotation being made so that the reading can continue after this point.

[2] *Review&Go* supports annotations to be hosted locally in reviewers' browser or at the *Hypothes.is* web service

Account for familiarity: *Review&Go* gestures as *Acrobat Reader*

This requirement aims to preserve *Acrobat Reader*'s gestures for highlighting, commenting, and overviewing. The next paragraphs abound on how this non-functional requirement impacted *Review&Go*'s design. **Highlighting**. *Acrobat Reader* supports two modes. Sporadic mode: select the target paragraph; next, click the highlight button. Continuous mode: select the highlight button; next, keep selecting paragraphs that are readily highlighted. Likewise, *Review&Go* supports two modes: the continuous mode for quality characteristics (i.e., rigor, design, and relevance), and the sporadic mode for the attributes.

Commenting. Once on a highlighted paragraph, *Acrobat Reader* adds comments by double-clicking. So does *Review&Go*. The difference stems from the comment canvas. *Acrobat Reader* holds the date, author, and the text. *Review&Go*'s supports the graduation, the comment itself, and the reference finder box.

Overviewing. *Acrobat Reader* offers a list-of-comment tab that behaves like an index: click on one of the comments to move to the document's paragraph where this comment was made. The *Review&Go* counterpart is the canvas (see Fig. 5.6). But the canvas is not just an index. It is intended to offer a quick glimpse of the manuscript's merits by clustering annotations by reviewing criteria. In so doing, it promotes a glance at the strengths (green background) and limitations (red or yellow background) of the manuscript.

5.4.2 Intervention and evaluation

This section reports on the intervention and evaluation of the proposed customized annotation tool for peer review, *Review&Go*. The evaluation was planned through the GQM (Goal, Question, Metric) paradigm [BCR94].

Goal. The purpose of this study is

> to *predict* the adoption of *dedicated highlighters* for improving *reviewers' efficiency* and *review effectiveness* from the point of view of *reviewers* in the context of a *conference manuscript revision*.

Questions. To better profile what we mean by "predict", we resorted to a reduction of Roger's model of Diffusion of Innovations that includes only those constructs consistently related to the Technology Adoption Model (TAM): relative advantage, complexity, and compatibility [TK82]. Specifically, three general questions are posed:

- Is *Review&Go* perceived to be better than conducting the review through *Acrobat Reader*? (Relative Advantage)
- Is *Review&Go* perceived to be consistent with the existing values, needs, and past experiences of *Acrobat Reader*? (Compatibility)
- Is *Review&Go* perceived to be difficult to use? (Complexity)

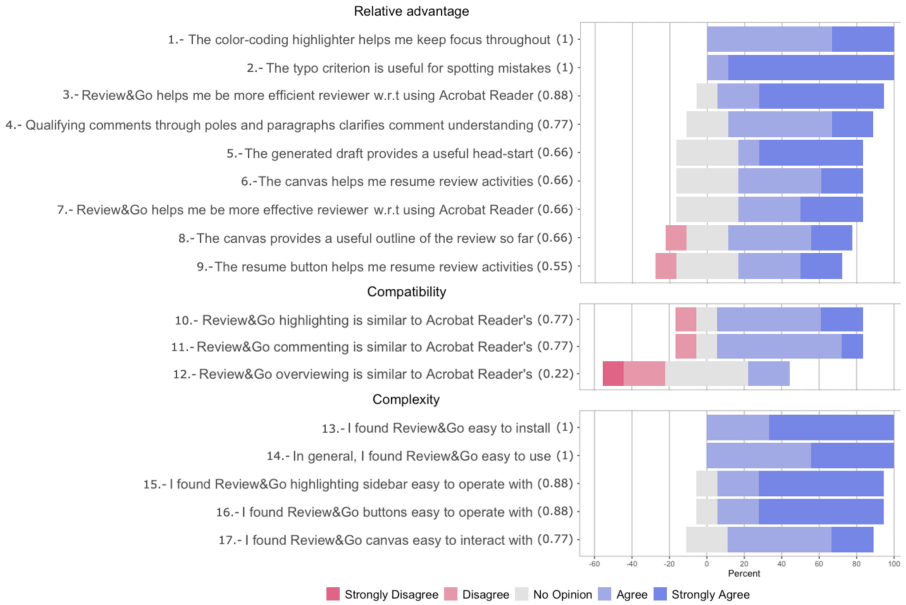


Figure 5.7: Diverging Stacked Bar Chart for the Perceived-Adoption Questionnaire using a 5 point Likert scale.

Metrics. Each of these questions is next refined in terms of the Design Principles that guide *Review&Go* (see Fig. 5.7): relative advantage (questions 1 to 9), compatibility (questions 10 to 12) and complexity (questions 13 to 17). The Cronbach's alpha values of the three dimensions were 0.77, 0.71, and 0.71, implying acceptable reliability of the questionnaire. Finally, metrics are derived from the participants' answers as normalization of good perception (i.e., Agree, Strongly Agree) vs. the total number of answers. Hence, "1" will stand for the highest perception. Next, we provide details of the evaluation.

Participants identification. Participants were recruited locally. All of them are part of the lecturers' board from the faculty of computer science at the University of the Basque Country. For participants to qualify as reviewers, they should have experience in reviewing papers (minimum of two), and specifically, knowledge of DSR methodology. Six lecturers and three post-graduate students qualified. Participants were given a ten-minute introduction to *Review&Go* where a sample manuscript was reviewed.

Methodology and data collection. Participants were asked to review a paper from previous editions of DESRIST. Papers were selected based on claiming the use of DSR as the research methodology. To check out resumption utilities, the revision was interrupted for 20 minutes so that the short-term memory was reset. Once the testing session was over, participants were asked to fill in a questionnaire that rates different aspects of *Review&Go* along a five-point

Likert scale (see Fig. 5.7). The questionnaire builds upon constructs consistently related to technology adoption behavior: relative advantage, complexity, and compatibility [McE04]. In addition, open comments were also welcome.

Results. Fig. 5.7 outlines results using a Diverging Stacked Bar Chart. These charts are recommended where the primary interest is in the total count (or percent) to the right or left of the neutral answer (i.e., "No Opinion"). The breakdown into strongly or not is of lesser interest so that the primary comparisons do have a common baseline of zero. The resulting metrics are added at the end of each question along with the formula: (#Agree + #StronglyAgree) / #Participants.

Discussion. In general, users perceive *Review&Go* as providing a relative advantage with regard to using *Acrobat Reader* highlight facilities (questions 3 and 7 in Fig. 5.7). Next, *Review&Go* gestures were considered quite consistent with those of *Acrobat Reader* (questions 10 to 12) except the way of obtaining overviews. Some additional comments follow.

First, highlights are often used. It is a way to pinpoint meaningful paragraphs. Yet, participants felt a bit overwhelmed with the 15 quality attributes offered as a default. Some of them preferred to focus first on the main quality categories (i.e., relevance, design, and rigor), and next, move down to the measurement attributes in subsequent readings, if necessary. In the same vein, five participants introduced their quality criteria (e.g., "understandability"). This suggests that customizability is certainly a must for color-coded highlighters in the peer-review context. A participant suggested "review criteria cartridges" that, provided by journals and conferences, could automatically configure the highlighter.

Second, surprisingly, the "typo" button to effortlessly report misprints was the highest in rank. Participants also appreciated color-coded and the draft generator as a transcript utility (no need for manual copy&paste) but also as a way to have comments arranged along with grades. Two participants observed the inability to introduce comments without associated highlighted paragraphs. This prevents general observations from being captured if no related paragraph exists.

Third, the canvas received neutral punctuation. Participants appreciated its role as an index on top of the manuscript highlights but with not so much enthusiasm. Its role as a resumption utility was not appreciated, more likely due to failure in re-creating a realistic scenario where the manuscript size and real evaluation needs would lead to longer reviewing times, hence, making more compelling the need for resumption support. One participant observed the interest in the canvas for article documentation as a sort of bibliographic record.

This moves us to the second BIE cycle.

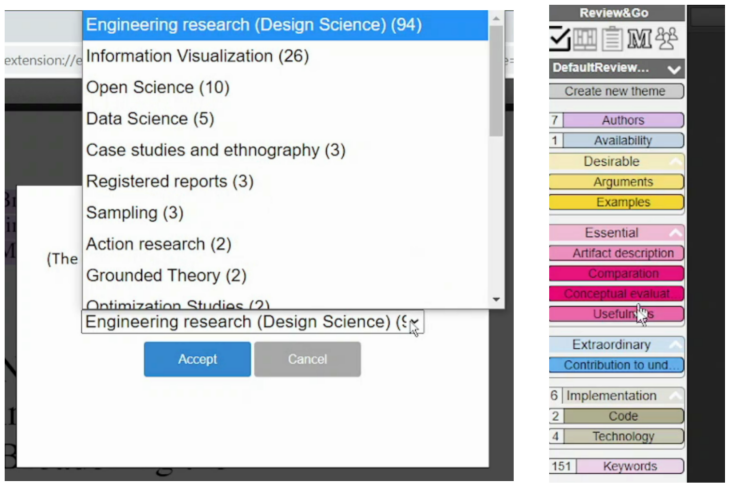


Figure 5.8: A) Selection of empirical standard checklist. Note the number in parenthesis for each category represents the number of keywords related to this topic have been found in the paper. B) Resultant highlighter orders evaluation attributes by desirable, essential and extraordinary.

5.5 BIE2: Acting upon review quality based on Empirical Standards

Looking at the results of the preliminary evaluation, a list of improvements was proposed to improve *Review&Go*. This section takes those results as the starting point for a new iteration. At the point of this writing, we are in the middle of the second cycle in the building phase. Therefore, an evaluation is still pending to evaluate improvements presented in this section and confirm the previous results in a real setting.

5.5.1 Build

In the second building iteration of the *Review&Go* project, we act upon contextualization and timely reviews. We start by re-building the artifact for contextualized reviews.

Improving contextualization: Empirical standards

Review&Go evaluation participants addressed the color-coded highlighter as the most positive aspect of the tool. However, one of the problems of *Review&Go* found is that the default review framework provided could not be useful to review papers from other venues than a paper from the DESRIST conference used in the evaluation. In that way, they mentioned that defining or customizing a review

framework is not that easy. In the same way, as mentioned in the problem formulation (see Section 5.2) what is understood as a quality standard (i.e., manuscripts with enough quality to be published) by different reviewers can be different, but also, depending on the venue or research area can be different too. To solve this, the Association for Computing Machinery (ACM) has created the Empirical Standard. The ACM SIGSOFT Empirical Standard [RBB+20] is a brief public document that communicates expectations for empirical research in Software Engineering.

The Empirical Standard tries to mitigate problems in peer review by providing guidelines to let reviewers question the soundness of the paper in review. They provide different guidelines, depending on the type of research method the authors used, and checklists for each of the methods to validate attributes of what is expected in each type of research method. Those attributes can be essential (i.e., necessary conditions for publishing the work), desirable (i.e., attributes that are recommended but not always are necessary or applicable), and extraordinary (i.e., attributes that are not required even in the most prestigious or demanding venues).

Currently, Empirical Standard provides a general checklist for all manuscripts, a checklist for engineering research (e.g., DSR), qualitative (e.g., action research or grounded theory) and quantitative evaluation methods (experiments or questionnaires), and supplements (e.g., evaluate specific aspects such as open science, replicability, information visualization, among others). Each of the checklists provides essential, desirable, and extraordinary attributes that reviewers should measure.

Review&Go has been modified to support the configuration of these checklists. The checklist attributes work as a highlighter where the reviewers use color-coded annotations to pinpoint evidence found to check (or uncheck) each of the attributes (see Fig. 5.8). In the same way, a new view is provided in the canvas for summarizing and decision making (see Fig. 5.9).

Improving timeliness: Keyword lookup

Related to the previous mechanism, checklists can be provided by the venue[3] or can be defined by the reviewers themselves. In the second case, it would take a bit of time to guess from the paper the conducted research type. To this end, *Review&Go* has included a feature for automatic keyword lookup. It analyzes the whole manuscript's text looking for words that are recurrently used in some research methods (e.g., qualitative surveys can be identified by searching for keywords such as "semi-structured interviews" or "synchronous conversation", while papers following a systematic review method usually use "primary study" or "search string")[4]. In this way, *Review&Go* suggests evaluation checklists based on the number of occurrences of keywords found (see Fig. 5.8). Additionally, as keywords are also annotated in the manuscript, those annotated keywords work

[3] Checklists provided by EASE'21: https://easychair.org/cfp/EASE2021

[4] A complete list of keywords matching each of the research methods can be found here: https://rebrand.ly/empiricalStandardKeywords

Figure 5.9: The canvas supports navigation through essential, desirable and extraordinary attributes and for each of them list the annotations (evidence) supporting the attribute decision.

as hints for reviewers to look for possible evidence for a specific criterion in the checklist.

5.5.2 Intervention and evaluation

As mentioned before, at the point of this writing, we are in the middle of the second cycle in the building phase requiring an additional evaluation of *Review&Go* in a realistic setting (e.g., conference or journal). In this respect, approaching PC chairs is on the radar. PC chairs have the most interest in improving reviews for attracting submissions and enhancing authors' satisfaction. By providing a "review-criteria cartridge", PC chairs can tune *Review&Go*'s color codes, facilitating review harmonization and, hence, pooling sessions.

5.6 Formalization of Learning

ADR intervenes and improves a specific setting through a cycle of making changes, observing the resulting situation, and making further changes. We are "experimenting" by making adjustments and observing the effects of those adjustments in peer-reviewing. This improves validity but hinders generalizability and precision. As an ADR project, it should include a reflection of the extent of these threats. In this section, we discuss the generalization of the problem instance (i.e., to what extent the efficiency and effectiveness in peer review is a problem for other departments than ours compared with existing approaches), generalization of the solution instance (i.e., to what extent is *Review&Go* a solution for the problem) and derivation of design principles (i.e., what sort of knowledge can be distilled from the *Review&Go* project) [SHP+11].

5.6.1 Generalization of the problem

Peer review limitations have been addressed with a revolutionary or evolutionary perspective in different areas (see Table 5.1). Revolutionary approaches include rewarding with a kind of "currency" necessary to pay for getting their submissions reviewed [VN14, FP10] or where a cryptocurrency is used to recognize good feedback by authors, editors or readers [TFPSRH21]. Alternatively, evolutionary approaches do not change current practices but provide support to them: interoperability and transparency approaches look for increasing review quality by providing more fair and traceable reviews [Nos17, Nay16], tools to support collaboration among reviewers and editors to speed up reviewing process [EpA17], training programs for young scientists look for timely reviews increasing the number of reviewers [Cla10, GMC+15] and standardization and guidance facilitate decision making reducing bias [RBB+20].

Our approach is evolutionary, focusing on guidance and training while keeping interoperability. Specifically, we look for a dedicated W3C compliant annotation tool (increasing interoperability), while providing guidance supporting customized highlighters and ACM Sigsoft Empirical Standards. In that way, the

Table 5.1: Proposals for addressing peer review.

Measure	**Intervention**	**Outcome**	**Limitation**
Publons [VN14]	Enable reviewers to get credit from reviewing	Improved recognition	Dependent on funding agencies' use of data
PubCreds [FP10]	Reward reviewing activity with a currency necessary for getting submissions reviewed	Incentivizes peer review	Financial and organizational problems
Decentralized science [TFPSRH21]	Blockchain-based infrastructure for verification and recognition	Improved interoperability and recognition	Increases time spent
Open peer review [Nos17]	Inject transparency at different stages of the peer review process	Increased transparency & quality	Increases time spent & decline rate
Cascading submissions [GRAJ14]	Transfer rejected papers to partner journals conserving reviews	Decreased wastage of reviewer effort	Problematic between different publishers
Reviewer training [GMC+15]	Train young scientists in the peer review process	Increased number of reviewers	Inconclusive evidence of its effectiveness
Empirical Standards [RBB+20]	Provide guidelines and checklists for reviewing	Increased transparency & standardization	Increases time spent
EJPress / Hypothes.is [EpA17]	Collaborative web annotation tool	Facilitates dialog between authors, reviewers & editors	Privacy issues
Paperhive [Nay16]	W3C Annotation tool supporting comments	Facilitates interoperability	Lack of guidance in reviewing

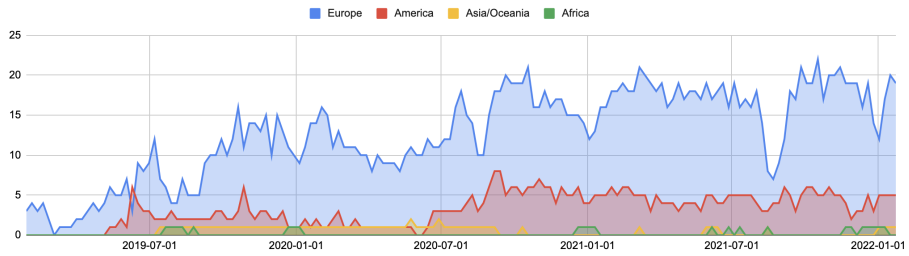


Figure 5.10: *Review&Go*'s user base from its release in early 2019 in Chrome Web Store distributed by region (source: Google Chrome Store).

addressed problem (i.e., efficiency and effectiveness in peer review) is recurrent in peer review literature overall.

5.6.2 Generalization of the solution

We now address the extent to which *Review&Go* might be a *solution* to efficient and effective peer-reviewing for stakeholders other than researchers at our university. To this end, we disclosed *Review&Go* to the public at the end of 2018. Specifically, we uploaded *Review&Go*, a video, and a user manual to the Chrome Web Store, and provided a demo session at PEERE'20. As for January 2022 (see Fig. 5.10), *Review&Go* is enjoyed by almost 26 users.

Software installation is regarded as a proxy for utility. It can be argued that the discretionary effort of installing *Review&Go* provides evidence of enough 'perceived utility' that it is at least of interest. Moreover, the fact that the number of users has remained steady for more than two years points to sustained interest as evidence of utility. However, the main threat to construct validity of this evidence is to interpret 'installation' as 'real use'. It might be the case of users keeping *Review&Go* installed without really using it. Note also that factors from TAM, UTAUT, and other models of technology adoption and acceptance, such as perceived usefulness and ease of use, performance expectation, or effort expectancy, may be significant influencers of the discretionary effort to install *Review&Go*. As mentioned in Section 5.4.2 an evaluation in a realistic setting is pending to increase the soundness of our solution.

Derivation of design principles

So far, we look at *Review&Go* as a whole. Now, we disentangle the distinct mechanisms that on balance are responsible for the usefulness and perceived ease of use as detailed in Section 5.4.2. Table 5.3 outlines the main design principles. Design principles reflect knowledge of both IT and human behavior [GKS20]. Accordingly, a design principle should provide cues about the effect on the target audience, the technological cause, and how it is instantiated in the solution. Table 5.2 outlines our proposal for peer review efficiency and

Table 5.2: *Review&Go* as an instantiation of the peer-review Design Model.

Expected Effect on researchers behavior	Technological Cause	*Review&Go* instantiation
specific and contextualized review	color-coded highlighters	empirical-standard-based review framework
selective review	supplemented comments by "gradations" (minor vs major concern)	gradable comments and literature reference
facilitate decision-tacking	review summarization weighting merits and demerits	canvas view
reduce report writing time	provide a head-start template for review reporting	draft generation out from annotations
facilitate resumption of review activity	support for interruption resumption	back-to-last annotation
facilitate use	familiarity with currently used tools	preservation of Acrobat Reader gestures

effectiveness and its instantiation in *Review&Go*.

5.7 Conclusion

The peer-review system is under pressure, partially due to an increase in the number of submissions. To improve reviewers' productivity, we advocate moving beyond Acrobat-Reader-like facilities to dedicated highlighters that account for review specifics. We introduce meta-requirements for dedicated highlighters. These requirements are being formatively tested out through *Review&Go*. Results are certainly promising but far from being conclusive. It should be noted that the sole reliance on subjective measures is a limitation of our study.

Next follow-on is to evaluate *Review&Go* in realistic settings. In this respect, approaching PC chairs is on the radar. PC chairs have the most interest in improving reviews for attracting submissions and enhancing authors' satisfaction. By providing a "review-criteria cartridge", PC chairs can tune *Review&Go*'s color codes, facilitating review harmonization and, hence, pooling sessions.

Part of this chapter has already been published at DESRIST'19 and PEERE'20 conferences:

- Díaz, O., Contell, J. P., & Medina, H. (2019). Performant Peer Review for Design Science Manuscripts: A Pilot Study on Dedicated Highlighters. Lecture Notes in Computer Science (Including Subseries Lecture

Table 5.3: Mapping *Review&Go* project to ADR principles.

ADR principles	The ADR process in the *Review&Go* project	Main Actions
Principle 1: Practice-Inspired Research	Research was driven by the need to provide quality feedback in peer-review	Investigated current practice and needs by researchers at our research group
Principle 2: Theory-Ingrained Artifact	This research project is informed by theories in social-science for quality feedback and practices in peer-review	Revision on peer-review literature
Principle 3: Reciprocal Shaping	Researchers & Web Engineers teamed up to scope and shape the intervention	*Review&Go* unfolded through prototyping creating a Minimum Viable Product (MVP)
Principle 4: Mutually Influential Roles	The ADR team included three Web Engineers, one evaluation specialist and up to 9 researchers	Sharing meeting to consider technical feasibility of researchers' enhancement suggestions
Principle 5: Authentic and Concurrent Evaluation	An evaluation has been conducted with 9 researchers. Feedback from real users has been received as the extension is published in the Chrome Web Store.	An evaluation was conducted with 9 researchers to measure relative advantage, compatibility and complexity of *Review&Go* using a TAM questionnaire
Principle 6: Guided Emergence	The preliminary design of *Review&Go* was continuously reshaped through use and feedback from researchers	An evaluation was conducted where three main insights where provided (see Section 5.4.2)
Principle 7: Generalized Outcomes	A set of design principles for peer-reviewing was articulated	*Review&Go* was made publicly available through the Chrome Web Store

Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 11491 LNCS(1), 61–75. DESRIST'19. **CORE A, Class 3**.

- Diaz, O., Contell, J. P., & Medina, H. (2020). Software scaffolds for quality feedback in peer review. PEERE'20, 1–4.

Chapter 6

Promoting Design Knowledge Accumulation Through Systematic Reuse: The Case for Product Line Engineering

6.1 Introduction

In the previous chapters, we have introduced three practices where problems in efficiency and effectiveness have been faced using customized annotation tools. For example, the *Highlight&Go* project increases efficiency in data extraction while keeping quality attributes for extracted data (consistency, traceability, completeness, and observability). In the case of *Mark&Go* and *Review&Go* provided feedback also addresses efficiency in terms of timing, while increasing other quality attributes that this kind of feedback requires (to be specific, personal, or contextualized). To address each problem, each artifact has considered different mechanisms that have enabled it to fulfill the requirements of the specific domain. For example, to provide timely feedback in *Review&Go* we have developed the head-start template generation mechanism, which automatically provides a review template from created annotations to facilitate the writing of the review report.

In our case, the design considerations in *Mark&Go* contributed to the knowledge base that is used to inform the design considerations in *Review&Go*. For example, contextualized feedback in *Mark&Go* is solved by color-coded annotations based on an evaluation rubric. This advance in the knowledge base informs *Review&Go*, where contextualized reviews should be addressed. To enable contextualized reviews in peer review, color-coding annotations can be reused to improve contextualized feedback in peer review, but, in this case, based on a review framework (i.e., Empirical Standard).

From a broad perspective, Design Science Research, and consequently Action Design Research, aims to come up with Design Knowledge (DK), i.e., means-end relationships between problem and solution space [Ven06]. In a recent report, Vom Brocke et al. regret that "most studies focus on a single DSR project, aiming at deriving DK within this project, while knowledge accumulation and evolution across projects are rarely considered as an antecedent or contribution of the project" [VWHM20]. It might be partially due to the limited reuse of design artifacts. Unlike commercial artifacts, in research software, like the designed web annotation tools, design artifacts are not an end in themselves but a means to advance DK.

Here, artifacts are the carriers of DK's mechanisms, i.e., the means that "either lead to or allow users ... to accomplish an aim" [GKS20]. Mechanisms are the way through which DK aims to impact a relevant problem. If DK is essentially evolutionary, so should it be the underlying artifact as the necessary bearers of DK's mechanisms. Accordingly, if the artifacts are "*rigid*", then the underlying DK will be more difficult to be accumulated by other DSR projects. This makes previous authors distinguish between *fitness-for-use* (i.e., the ability of the design artifact to perform in the current application context with the current set of goals in the problem space) from *fitness-for-evolution* (i.e., the ability of the solution to adapt to changes in the problem space over time). This distinction was enshrined by Gill and Hevner when positing that "the evolutionary fitness of a design artifact is more valuable than its immediate usefulness" [GH13b].

This situation arises especially in software artifacts, and web annotation software is not far from this phenomenon. Studies, where the annotation is used as a medium or as a goal, are limited to the reuse of already designed artifacts (i.e., annotation systems), and consequently, developers and researchers are reimplementing DK's mechanisms (i.e., annotation functionalities) instead of deriving them from previous annotation projects.

When it comes to software development, two common practices might jeopardize *fitness-for-evolution*. First, researchers and developers often make suboptimal developments to get to the evaluation quickly, get feedback, and gain *fitness-for-use*. Speeding up time-to-evaluate might well play the role of time-to-market in the commercial world. Here, suboptimal development decisions lead to the so-called *technical debt*, i.e., the accumulated backlog of software development is needed because developers favor a quick solution over a "fitter solution", usually to reduce implementation time [SSK19]. Second, artifact reuse among research projects is frequently achieved via *clone&own*. Here, a new product starts by cloning an existing one, and then developers adapt parts of it to meet the new requirements. This is the case of some web annotation tools such as EJournalPress [EpA17], Digipo [Cau20], QDR [EK18] or FakeNewsAnnotationTool [RMSB19], just to mention a few. Although cost-saving in the short run, *clone&own* is hardly scalable if the track about the mechanisms existing in several clones is lacking [FMS+17]. As a result, *clone&own* increases 'DK entropy': different projects P1, P2,..., Pn might explore nearby design regions adding eventually new stakeholders, goodness criteria, or eval-

uation settings, but conducted upon distinct artifacts: A1, A2,..., An. These artifacts are similar insofar as they might be obtained through cloning, yet their code mechanisms are dispersed as their variations are difficult to trace and compare. In short, technical debt together with *clone&own* practices might lead to *design debt*, i.e., deferring a holistic understanding of the underlying design principles.

If we draw parallels with manuscripts, SLRs follow protocols to ensure a *systematic* approach to knowledge accumulation. In the same vein, if research software artifacts can be used as DK accumulators, software reuse should also be *systematic* to facilitate DK accumulation. This turns A1, A2,..., An from being independent products to becoming a *product family*. Development wise, this notion of "family" implies a clear distinction between Domain Engineering (i.e., where the platform is handled through "development for reuse") and Application Engineering (i.e., where specific products are derived from the common platform with a focus on "development by reuse") [ABKS13]. In other words, design artifacts are handled as a portfolio of related products using a shared platform and an efficient means of production, i.e., using Software Product Line Engineering [PBvdL05]. The Software Product Line Engineering (SPLE) development methodology advocates for systematic reuse by putting the focus on a family of artifacts rather than on one-off artifacts. As a consequence, the resulting SPL, which defines a common platform to derive annotation tools, will reduce the development effort for subsequent tools made of it.

In this chapter, we present the efforts to adopt SPLE to explore the accumulation and evolution of DK along with the three annotation projects presented in previous chapters. Specifically, this chapter tunes SPLE for Design Science Research software artifacts, specifically by:

- introducing a design process to achieve *fitness-for-evolution* and *fitness-for-use*
- operationalizing this process along SPLE that will be used as a guide to solving the problem addressed in this thesis with three annotation use cases and the generalization of the solution

6.2 Background

Fit artifacts allow for reuse and extension to settings other than those originally contemplated, hence increasing projectability (e.g., broader applicability scope) and confidence (e.g., sounder evaluation) [VWHM20]. For processes [Win12] and conceptual models [VBB06] reuse has been already addressed. For software, frameworks and configuration have been proposed [MT04]. In the annotation artifacts development realm, AnnotatorJS [pro15] and HUMAN [WRD+20] are examples of frameworks and configuration, respectively.

However, frameworks and configurations do not fully account for artifact reuse. First, *frameworks* limit extensions to actuate upon the hot spots foreseen by the framework [MMM98]. This might compromise future requirements

that might not be accommodated through the envisioned hot spots. Alternatively, *configuration* captures variations within if-then statements where "if" conditions are evaluated at run time against a configuration file. Here, a single artifact handles all functionality, no matter whether a configuration option will never be selected in a certain scenario by its users. From a DSR perspective, configuration hinders exploration of *separated* artifacts rather than a *single* one with all possible configurations built in. The single approach needs to check configuration dependencies for different design criteria, requiring additional code development to check for run-time defined configuration conditions.

Alternatively, the configuration can be moved from run time to compile time through *Conditional Compilation*. Here, variants are enclosed within *#ifdef* and *#endif* marks, and associated with precompilation directives, i.e., Boolean expressions upon "configuration parameters". The important point is that so-marked code is conditionally removed before compilation. The *cpp* preprocessor is a case in point [ANS03]. Conditional compilation just delivers the code that is needed for the selected variant. And, what is also relevant, configuration-dependency checking is outsourced from the application code to dedicated configurators. Yet, reusability is not only a matter of programming effort but of being *systematic*, i.e., identifying, understanding, and managing the set of processes and roles that interplay in making software reusable. This is when SPLE comes into play.

SPLE aims to identify commonality and variability among applications *within a domain*, and build reusable assets to benefit future development efforts [PBvdL05]. In SPLE, the product plays an ancillary role in favor of the notion of "domain" (e.g., web annotation tools). The result is an SPL, i.e., "a set of applications sharing a common, managed set of functionalities that satisfy the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way" [CN02].

DSR wise, we rephrase this definition by describing an SPL as *a set of design artifacts sharing a common managed set of DK's mechanisms that satisfy the specific needs of a particular design region and that are developed in a prescribed way*. In the next sections, we will introduce what sort of prescription (Section 6.3) and how to conduct the fitness cycle in SPLE (Section 6.5).

6.3 Fit design as a Continuous Improvement practice

'Fitness' departs from "utility" [GH13b]. The utility might be evaluated with regard to current conditions. By contrast, "fitness" is (partially) evaluated regarding foreseen conditions. The current conditions are frequently difficult to be accurately apprehended, yet alone estimations about future requirements. We recognize this difficulty in coming up with fit artifacts, and hence, put the focus on the *process* that might lead to fit artifacts. We might ignore what a "fit artifact" is, but we can provide the means and appreciate the practices

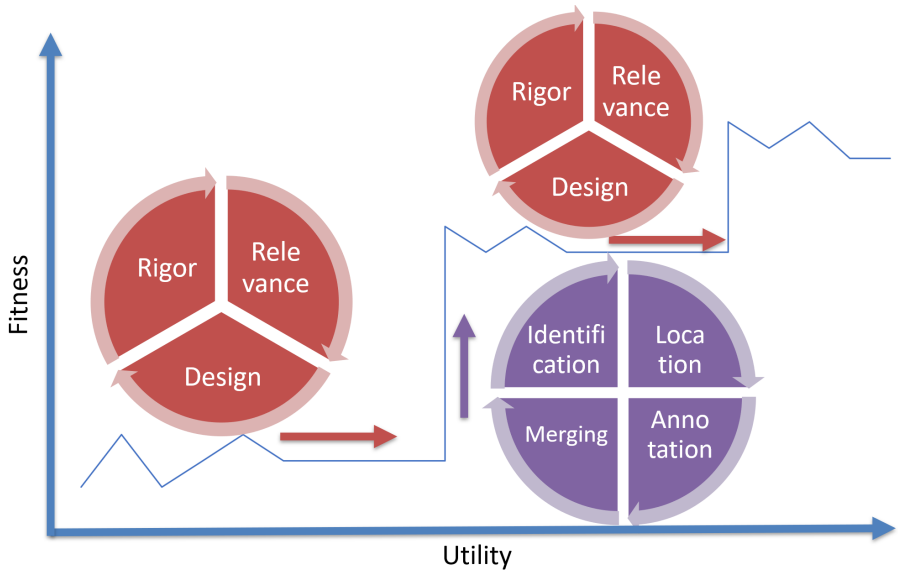


Figure 6.1: Fit-minded design processes.

that increase the chances to come up with fit artifacts. This vindicates a shift from *the result* (i.e., the artifact) to *the process* (i.e., the design). This is when Continuous Improvement comes into play [Mas86].

Continuous Improvement (CI) was born as a management practice for organizations that need to be fit, i.e., by promptly responding to changing customer needs, market changes, and competitive threats. This description bears resemblance to DSR challenges: "both the problem and solution spaces are subject to constant and increasing change, so that the past DK is prone to rapid aging ... and, hence, DK requires constant updates in the form of revision and further evolutionary development" [VWHM20].

Based on this resemblance, we can look at CI in the search for fit artifacts. Specifically, CI sustains that the desired result is achieved more effectively when related resources and activities are managed as a process [SS15]. This process commonly intermingles two loops: the improvement cycle and the standardizing cycle [Mas86]. The former goes along the "plan-do-check-act" (PDCA) cycle. *Plan* refers to setting a target for improvement. *Do* means implementing the plan. *Check* is the control for the effective performance of the plan. Finally, *Act* refers to standardizing the new (improved) process and setting targets for a new improvement cycle. As the resulting work process, following each cycle of improvement, becomes unstable due to the nature of change, a second cycle is, therefore, required to stabilize it: the "standardizing cycle" that goes along the "standardize-do-check-act" (SDCA) cycle. The main purpose of this cycle is "to iron out abnormalities in the resulting work process and bring it back to harmony before moving to a new improving cycle" [SS15].

Accordingly, we advocate for a similar approach to *fit-minded design processes* (see Fig. 6.1):

- the Utility Cycle (i.e., the counterpart of the PDCA cycle), which stands for the activities associated with relevance-design-rigor [Hev07],
- the Fitness Cycle (i.e., the counterpart of the SDCA cycle) which complements the previous one with "refactoring" activities, i.e., tuning the artifact to be eventually reused, adapted, or appropriated by scenarios/developers other than those originally considered.

Broadly, the Utility Cycle ends with an evaluation of the utility of the artifact and some design principles that are abstracted out of the experience (i.e., *fitness-for-use*). At this point, the development team faces a crossroads. On the way towards DK accumulation, Vom Brocke et al. introduce the metaphor of a journey along with a three-dimensional space: *projectability* of the problem in the problem space, *fitness* of the solution in the solution space, and *confidence* in the current evaluation evidence. This journey is marked by the development of different DSR artifacts that explore distinct stakeholders, contexts, or related practices (i.e., a design region). Traditionally, these DSR artifacts tend to be kept separated where reuse is frequently conducted through *clone&own*.

We depart from this scenario in two ways. First, we advocate for transiting this three-dimension space through intertwining "utility cycles" (advancing projectability and confidence) and "fitness cycles" (advancing fitness). Second, this process is conducted not through *clone&own* but through systematic reuse (i.e., SPLE). At the onset, a first artifact *A* is engineered for variability (i.e., making the artifact flexible to ensure reuse), resulting in a fitter *A'*. By intertwining "utility cycles" and "fitness cycles", additional artifacts fleshed out distinct DK advances, resulting in a set of artifacts A', A", A"', etc. Rather than keeping each artifact apart, a platform is gradually generated in the fitness cycle. This platform collects commonality and variability in the design space in terms of variation points. This platform is the SPL. The main premise is that most developed design artifacts are not brand-new artifacts but rather variants of other artifacts within the same design region. Hence, and except for the very first iteration, DSR artifacts are obtained out of the SPL.

To be effective as an accumulation mechanism, SPLs should receive feedback from the insights gained in adapting the SPL artifacts to scenarios other than those initially considered by the SPLs. Artifact developers branch off the SPL codebase and adapt the core code to account for unexplored design regions. Once these regions have been explored "utility cycles", and the mechanisms have been accordingly adapted/created and evaluated, the "fitness cycle" cares about merging back these new developments into the main SPL branch (see Fig. 6.2 for an example of how the utility and fitness are reflected in a sample SPL, more details on section 6.6). The vision is then for *SPLs to embody DK that goes beyond a design artifact to include a set of artifacts, i.e., a product family, and, in so doing, facilitates DK accumulation for a given design region.* Next, we will introduce the pilot example in this thesis, *Highlight&Go*.

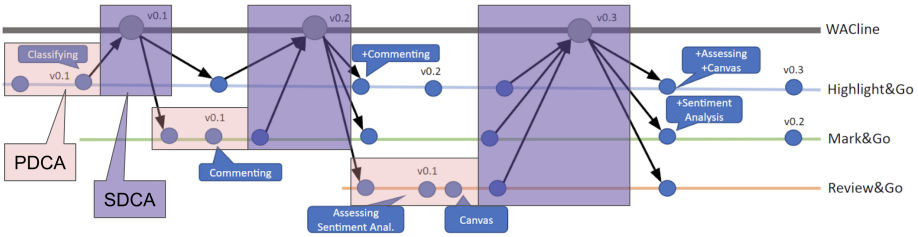


Figure 6.2: Branching model of *WACline* displaying some of the implemented mechanisms in PDCA cycle (in red) and SDCA cycle (in purple) and reuse of foreseen mechanisms in previously developed artifacts.

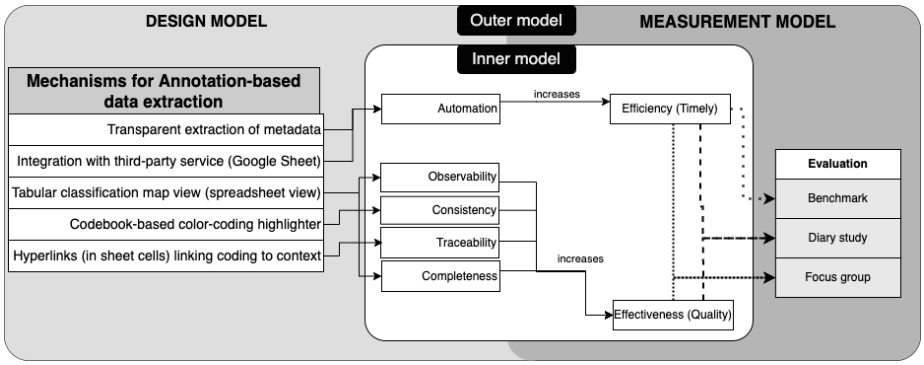


Figure 6.3: Inner-outer model of *Highlight&Go* project presented in Chapter 3. Mechanisms are renamed to abstract from specific naming used in SLR context and facilitate comprehension of the DK accumulation process.

6.4 Pilot study: *Highlight&Go*

Highlight&Go tackles the lack of software scaffolds for data extraction in systematic literature reviews and mapping studies. This project was conducted using ADR and presented in Chapter 3. The Fig. 6.3 shows the inner-outer model of the *Highlight&Go* project. On one hand, the *Inner Model* refers to the Justificatory Theory that introduces the variables to act upon, where five independent variables are introduced (automation, observability, consistency, traceability, and completeness) and two dependent variables are addressed (efficiency and effectiveness). On the other hand, the *Outer Model* describes how independent variables can be manipulated through an IT artifact (i.e., *Highlight&Go*), and how dependent variables are measured (e.g., focus group). The resulting DK is tentatively abstracted in terms of Design Principles and, consequently, implemented mechanisms.

Therefore, the evaluation is based on the notion of utility as usefulness (e.g., effectiveness and efficiency in performing data extraction). From this perspective, *Highlight&Go* is evaluated as useful. Yet, Gill et al. introduce an additional

utilitarian perspective, i.e., that of evolution [GH13b]. *Highlight&Go* can be deployed (reproduced) across researchers, but it can be useful to explore the design landscape of quality and efficiency beyond data extraction. This evolution can transit across two main dimensions: confidence (i.e., the extent of evaluation comprehensiveness, given the great variety of different methods and application scenarios) and projectability (i.e., the extent of the problem-analysis comprehensiveness in terms of the context that frames the software) [SF18, VWHM20].

The confidence dimension. *Highlight&Go* was evaluated as a whole. However, Niehaves et al. advise a more piecemeal approach to ascertain how each mechanism ponders the final result, and whether inter-dependencies of simultaneously implemented mechanisms might exist [NO16]. In this respect, we might be interested in calibrating whether effectiveness or efficiency has the strongest impact on *Highlight&Go* adoption. Even a single independent variable (e.g., automation) might be impacted by different mechanisms (e.g., transparent extraction of primary studies metadata and integration with Google Sheets).

The projectability dimension. As *Highlight&Go*'s insights can be very specific to the context of literature review data extraction, it is possible to increase projectability by introducing additional stakeholders or brand-new contexts where web annotation can increase effectiveness and efficiency.

The latter is the case of *Mark&Go* and *Review&Go* where lecturers and reviewers are introduced as new stakeholders and the context also has evolved to instructional feedback and peer-review, respectively. Some of the mechanisms implemented in *Highlight&Go* can be reused to support those practices. This calls for artifacts to be fitted for the journey ahead.

Highlight&Go at the very beginning was conceived as a monolithic application, where mechanisms were difficult to isolate. The projectability of *Highlight&Go* required to adapt it to the new settings (i.e., assignment marking and peer review). Our first attempt was to use the *clone&own* approach. This meant a different clone for each possible journey. Yet, this resulted in divergent software projects, missing opportunities to share code and insights about the phenomenon at hand. Some first attempts were made to try to componentize *Highlight&Go*, yet we utterly failed since most of the mechanisms could not be isolated as single components. Rather, the realization of DK mechanisms frequently crosscut different functional units (e.g., files, classes, methods), as they are tangled and scattered throughout the codebase.

In short, in the search for a quick time to evaluate the utility cycle, refactoring efforts are (un)intentionally postponed. Yet, this *technical debt* might never be paid if the mechanisms do not need to be reused. In other words, reusability efforts do not need to be conducted no matter the mechanism but just opportunistically for those mechanisms that might eventually pay off. Hence, the Fitness Cycle includes exploring fertile design regions and making informed decisions on what reusability efforts should be incurred or paid off, and when. The next section builds a case for operationalizing the Fitness Cycle as an SPLE endeavor.

6.5 The fitness cycle

The Utility Cycle ends up with a design artifact that fleshes out DK's mechanisms reckoned to help users. Yet, this is not the end of the story. We advocate for designers to go one step further and analyze the extent to which existing mechanisms can be reused in pursuit of exploring nearby design regions. To this end, we follow SPLE and its distinction between domain engineering and application engineering. Domain engineering supports the identification, location, and annotation steps of the SDCA cycle presented in Fig. 6.1, while application engineering covers the merging step.

Domain engineering

Domain Engineering is the process of analyzing the domain and developing the reuse platform. Here, we distinguish between the problem space and the solution space. The former takes the perspective of stakeholders and their problems, and views of the entire domain [ABKS13]. In a DSR setting, the domain stands for a practice where a practical problem arises whose solution is to be mediated through design artifacts. The results of domain analysis are documented in a *Feature Model*. In the SPLE literature, a "feature" stands for "a characteristic or end-user-visible behavior of a software system" [ABKS13]. For our purposes, however, we are not interested in all "end-user-visible behavior" but just those aspects that might have an impact on "utility", i.e., the DK's mechanisms. Therefore, we conceive *a feature as a description of a DK's mechanism.*

However, not all mechanisms necessarily become features, or at least, not right away. We previously observed that the Fitness Cycle includes exploring fertile design regions and making informed decisions on what reusability efforts should be incurred or paid off, and when. Turning a mechanism into a feature might involve a costly refactoring process that should be balanced against opportunities for this cost to pay off. Considerations to be pondered about include:

- current utility, i.e., how the mechanism ranked during the last evaluation of the artifact,
- foreseen utility, i.e., how the mechanism might serve to explore "the design fitness landscape",
- resource availability, including both development (technical skills) and evaluation (participants to tap into).

The foreseen utility might be ranged along with four possible values: *reusable* (i.e., the mechanism can be used as it is), *adaptable* (i.e., the mechanism might need some tuning), or *detrimental* (i.e., the mechanism might be harmful or users like not to have in the new setting). Only mechanisms ranked as *reusable* are moved unchanged during the refactoring process. They conform "the commonality" of the product line. By contrast, the rest of the options call for the mechanism's code counterpart to be customized or be removed altogether

```
// PVSCL:IFCOND(Metadacapture, LINE)
// Set metadata capture button
const metadataImageURL = chrome.extension.getURL('/images/metadata.png')
this.matadataImage = $(toolsetButtonTemplate.content.firstElementChild).clone
this.matadataImage.src = metadataImage
this.matadataImage.title = 'Capture paper metadata' // TODO i18n
this.toolsetBody.appendChild(this.matadataImage)
this.matadataImage.addEventListener('click', () => {
  this.captureMetadata()
})
// PVSCL:ENDCOND
```

Transparent Extraction Of Metadata

```
// PVSCL:IFCOND(GoogleSheet, LINE)
// Set Spreadsheet generation button
const googleSheetImageUrl = chrome.extension.getURL('/images/googleSheet.svg')
this.googleSheetImage = $(toolsetButtonTemplate.content.firstElementChild).cl
this.googleSheetImage.src = googleSheetImageUrl
this.googleSheetImage.title = 'Generate a spreadsheet with classified content'
this.toolsetBody.appendChild(this.googleSheetImage)
this.googleSheetImage.addEventListener('click', () => {
  GoogleSheetGenerator.generate()
})
```

Integration With Google Sheets

Figure 6.4: Preprocessor directives to annotate variant code for features third-party software integration in Google Sheet and transparent extraction of metadata.

to prevent features creeping in the new scenario. Variability-wise, the mechanism's code counterpart is a candidate to become a feature, i.e., amenable to be identified, annotated, and tested as a configurable option at precompilation time. After identifying candidate mechanisms we move to the solution space.

In the solution space design, implementation, validation, and verification of features realization and their combination to facilitate systematic reuse are covered [ABKS13]. This includes three steps:

- feature location refers to deciding which source code supports a given feature. Here two main difficulties arise. First, features tend to be rather domain-specific entities and orthogonal to typical structures found in programs, such as components, classes, or methods [ABKS13], so they are crosscut, scattered, and tangled along with the codebase. Second, features are rarely documented, developers' knowledge about the features fades quickly, and developers leave projects.

- feature annotation accounts for documenting the connection between a feature and its implementation. A common technique is the use of preprocessor directives (aka *#ifdef directives*) (see Fig. 6.4). Those directives resolve a Boolean expression upon a configuration of features (e.g., whether a feature is selected or discarded to be included in the resultant artifact).

Application engineering

During Application engineering, the needs of a specific DSR project are expressed in terms of existing DK mechanisms ready for reuse, i.e., features. Characterizing a DK's mechanism as a feature implies that artifacts can be derived in terms of these features (a.k.a. product configuration). Based on conditional compilation, *#ifdef* directives can be used to filter out optional code. Automation tools (e.g., Ant) are used to generate different artifacts based on the feature selection.

However, as mentioned before, existing features rarely fully satisfy the demands of the new DSR project. Features might need to be tuned while brand-new features might need to be introduced. That is, 'developing with reuse' provides a head-start, but it does not remove the need for artifact customization. Therefore, to be effective as an accumulation mechanism, SPLs should receive feedback from this customization, i.e., extend the initial 'domain' with scenarios other than those initially considered by the SPL. This feedback makes SPLs depart from *clone&own* insofar as customization branches do not persist dangling but end up being merged back to the reuse platform. It is this very merging effort that makes the SPL become the container for mechanisms that expand along a design region, i.e., the SPL domain (e.g., web annotation). It is from this perspective that we regard SPLs as the main enablers of DK accumulation. Now we move to how to instantiate the defined fitness process for web annotation tools.

6.6 From *Highlight&Go* to *WACline*

WACline tackles the heterogeneity of web annotation tools. *WACline* development follows the process presented in Fig. 6.5, where the *Highlight&Go* project was conducted and an artifact was implemented (PDCA cycle). At this point we started the fitness cycle that later let *Mark&Go* and *Review&Go* be benefited from this reuse.

6.6.1 Domain Engineering

We first analyze which *Highlight&Go*'s mechanisms are worth being turned into features in terms of current utility, available resources, and foreseen utility (see Table 6.1).

- current utility. *Highlight&Go* resorts to the evaluation presented in Chapter 3 where values 'High', 'Medium' and 'Low' are assigned based on how participants validated each of the mechanisms implemented.
- available resources. We focus on the availability of technical expertise. This might be an issue in academia, where artifact development mainly rests on the back of Ph.D. students. In our case, there was no problem as all the developers that had participated in the development of *Highlight&Go* were available.

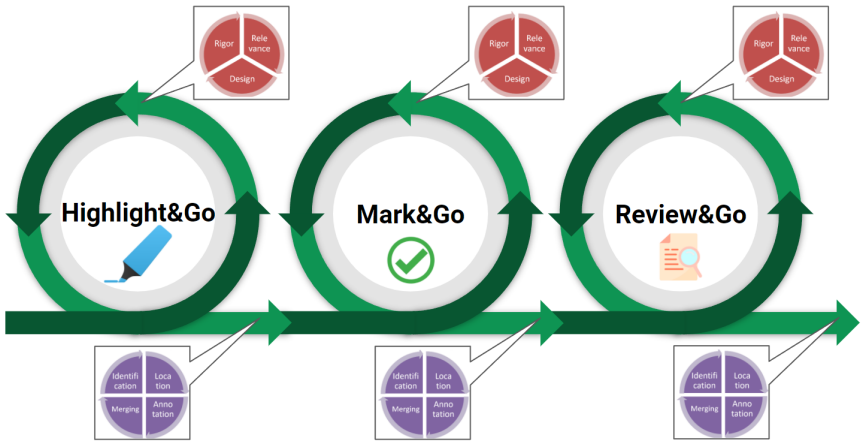


Figure 6.5: Design knowledge accumulation in developed project experiences are accumulative: the output of *Highlight&Go* is the departing point for *Mark&Go* and accumulation of *Highlight&Go* and *Mark&Go* outputs is the departing point of *Review&Go*.

- foreseen utility. Each mechanism is pondered for its potential utility in the following annotation practices. As reflected in Table 6.1, for the *Mark&Go* setting two features could be reused as they are, while the integration with a third-party service should be adapted (from Google Sheet to Moodle) and transparent extraction of metadata and tabular map view makes no sense in this context. To fully support requirements for *Mark&Go*'s project, some new mechanisms should be implemented.

At this time, we have considered the *Mark&Go* and *Review&Go* foreseen scenarios. This implied refactoring all the mechanisms as features keeping the commonalities as part of the core. This resulted in a 'fitter' *Highlight&Go*, the first release of *WACline* (see Fig. 6.2). In the case where all features of the first release of *WACline* are selected, then *Highlight&Go* is assembled back. Yet, some features might be deliberately left out (e.g., if they are not suitable for the new context or stakeholders).

6.6.2 Application engineering

Even at this very early stage, *WACline* accounts for varied artifacts. Different feature combinations result in distinct artifacts. Rather than developing by *clone&own* or starting from scratch, developers can now cherry-pick those features whose code is to be included in the onset codebase, branching off from *WACLine* (see Fig. 6.2). From then on, developers can customize the codebase to account for their scenarios' specifics along with the Utility or PDCA cycle. In this stage, application engineers, based on reused-as-is, adapted features and

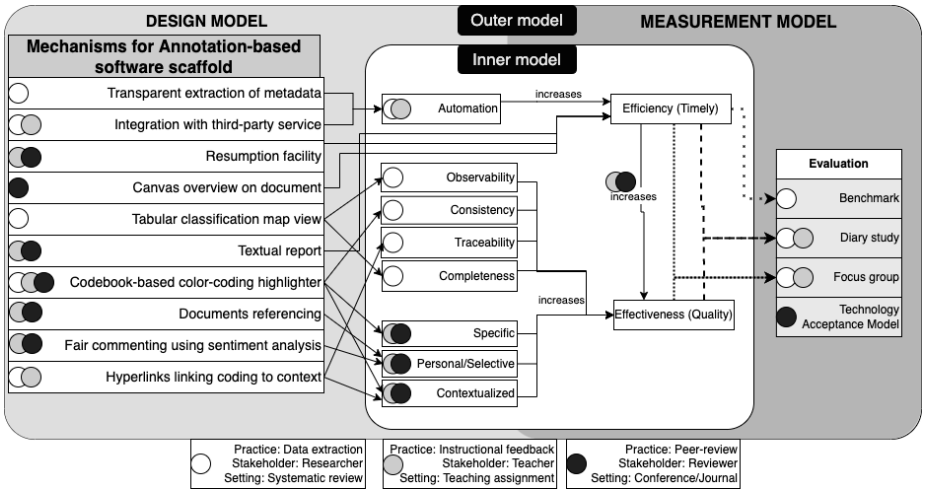


Figure 6.6: 'Accumulated' Inner/Outer Model: similar Inner Model where in all three approaches efficiency and effectiveness are measured but different Outer Models exist for the three different projects (bottom of the figure).

Table 6.1: Looking for feature candidates to be reused from *Highlight&Go* in consequent projects: *Mark&Go* and *Review&Go*.

Mechanism	**Current Utility**	**Resource Availability**	**Foreseen utility: *Mark&Go***	**Foreseen utility: *Review&Go***
Transparent extraction of metadata	Low	High	Detrimental	Detrimental
Integration with third-party service	High	Medium	Adaptable	Detrimental
Tabular classification map view	Medium	High	Detrimental	Adaptable
Codebook-based color-coding highlighter	High	High	Reusable	Reusable
Hyperlinks linking coding to context	High	High	Reusable	Detrimental

newly implemented features develop a second artifact. For our case, *Mark&Go* was implemented. Once the Utility Cycle for *Mark&Go* was over, a new Fitness Cycle starts, moving back to the fitness cycle where now identification, location, annotation, and merging are done over *Mark&Go*. For example, *Highlight&Go*'s hyperlinks linking to context were reused as it is, while modules implementing integration with Google Sheets were adapted for Moodle in *Mark&Go*. The same process is followed later by *Review&Go*. The resultant accumulated inner-outer model for the three different annotation projects hosted in *WACline* is presented in Fig. 6.6.

6.7 Conclusion

We make the case for SPLE to be incorporated during artifact development. Unlike *clone&own*, SPLE involves an additional refactoring and managerial effort. Yet, we conjecture benefits might go beyond those of software development (e.g., time-to-market, increase product quality, etc.) to include DK accumulation and evolution. Along with this chapter, we contribute by introducing and illustrating a fit-minded process where Utility Cycles intertwine with Fitness Cycles to produce an artifact: an SPL. An SPL accounts for a *set* of design artifacts that explore distinct features across a given design landscape (i.e., the SPL's domain).

The result of the process presented in this chapter works as a head-start point for the design of an SPL for web annotation client customization. In the following chapter, we will present a description and results obtained in terms of the development cost of reusing resultant SPL to face heterogeneity in web annotation clients.

Part of this chapter has already been published:

- Diaz, O., Medina, H., & Contell, J. P. (2021). Promoting Design Knowledge Accumulation Through Systematic Reuse: The Case for Product Line Engineering. In Proceedings of the 54th Hawaii International Conference on System Sciences. **CORE A, Class 1**.

Chapter 7

WACline: A Software Product Line for Web Annotation heterogeneity management

7.1 Introduction

In the previous chapter, we proposed to incorporate a fitness cycle into the Software Product Line Engineering process to facilitate the reuse of code among distinct ADR projects, promoting the design knowledge accumulation among them. We operationalize this process for the case of Web Annotation projects tackled in this thesis. The goal of this thesis is to facilitate the reusability of annotation tool mechanisms reducing their development and maintenance costs. To this end, we have designed and developed *WACline*. It is based on the resultant SPL from the design knowledge accumulation process conducted in the previous chapter, also known as a reactive approach [Kru01], to later refactor it based on the recommendations of W3C [W3C17] and a small analysis of the features already existent in third-party annotation tools.

This chapter presents *WACline* (Web Annotation Client-line) to face heterogeneity in annotation practices for review using an SPL architecture [ABKS13]. In the following lines, we describe the resultant SPL and the main results of its use in terms of code reuse for the three tools conforming to this thesis and the development of other three annotation prototypes by third-party developers to conduct annotation for concept mapping, bachelor's degrees thesis evaluation and legal sentences analysis, respectively.

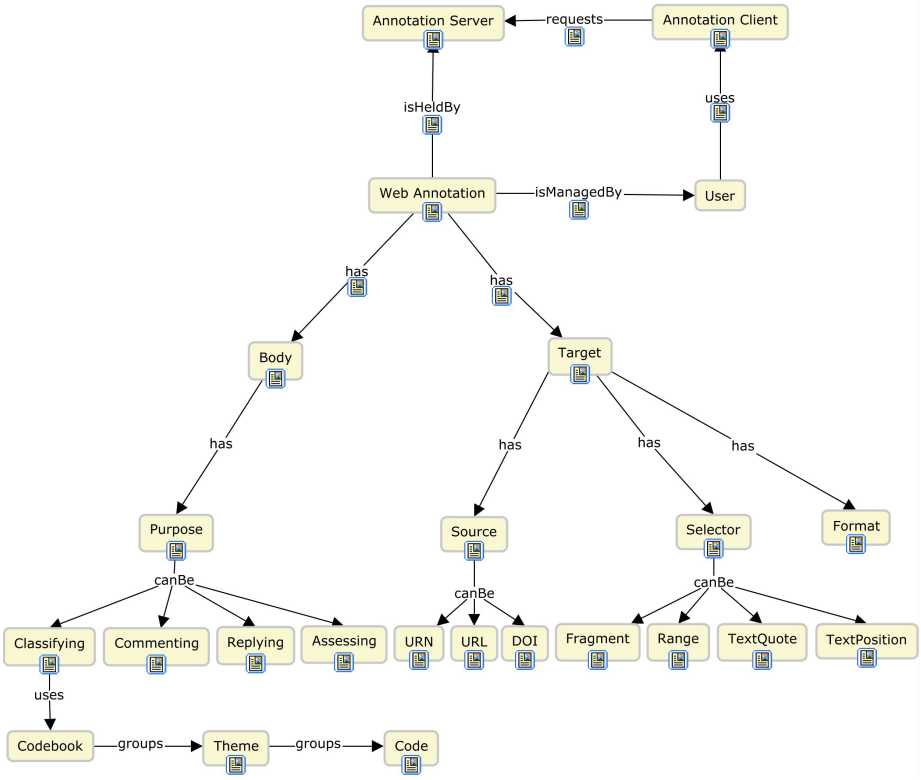


Figure 7.1: Concept map on Web Annotations. Map generated through *Concept&Go*. Interactive version available at https://rebrand.ly/webAnnotationCmap

7.2 The annotation model

In Section 2.2 we have introduced the W3C Web Annotation recommendations. These recommendations include the data model (describing the model and serialization), the vocabulary (underpinning classes, predicates, and named entities), and the protocol (describing mechanisms, annotation creation, and management). Fig. 7.1 shows a concept map. A concept map is a type of visualization to organize knowledge by defining concepts and relationships between those concepts. In this case, the concept map presents the main components captured in the W3C recommendation to describe web annotations and their relationships.

First, the Annotation Protocol introduces two constructs: the **Annotation Server**, where annotations are kept, and the **Annotation Client**, which serves as the conduit between the Annotation Server and the **Users**. Next, the Annotation Model introduces the notion of **Web Annotation** as a relationship between a **Body** (e.g., a comment) and a **Target** (e.g., a web resource or a

fragment of a web resource). A target is characterized by a triplet:

- **Source**, i.e the resource being annotated. It stands for an Internationalized Resource Identifier (IRI). Examples include URN, URL, or DOI,

- **Selector**, i.e., a locator that singles out the resource's segment of interest (e.g., highlighted). Selectors are used to unambiguously identify some regions/paragraphs of an image/document. Locators are fleshed out through distinct approaches: combining coordinates, a CSS, XPath expression, or the text's starting and ending positions. If Target has no selector, the annotated content is the whole document.

- **Format**, i.e., the way the document is realized. The same resource (IRI) can be rendered in multiple formats. For instance, a DOI identifies a manuscript no matter if it is delivered in PDF or HTML format in a Digital Library.

Finally, the annotation Body holds a **Specific Resource** (e.g., a comment or a classification code) that aims at a specific **Purpose**. W3C introduces thirteen different purposes [SCY17], where we have selected a subset of four different purposes that are of relevance for reviewing practices, but other new annotation practice variants may support new purposes:

- assessing: the purpose for when the user adds an annotation to assess or qualify in some way, rather than simply commenting on it, e.g., to write a review or assessment of a paper (e.g., qualify as weakness or strength in *Review&Go*) or validate or invalidate a previous annotation made by another person (e.g., data validation in data extraction in *Highlight&Go*)

- classifying: the purpose for when the user intends to classify the Target as something, e.g., to classify a text excerpt based on a codebook (in *Highlight&Go*) or rubric (in *Mark&Go*)

- commenting: purpose for when the user intends to comment about the Target, e.g., to provide a commentary about a particular text excerpt in a document to clarify the mistake done by a student in *Mark&Go*

- replying: the purpose for when the user intends to reply to a previous statement, either an Annotation or another resource, e.g., assisting (response) to solve a doubt about classifying a text excerpt annotation in *Highlight&Go*

W3C describes how annotation purposes can be instantiated in a data model (see Fig. 7.2), but does not preclude how these purposes are achieved. This is the labor of the annotation client, it has to provide a suitable interface to end-users. For example, *Hypothes.is* provides a user interface to let users annotate text excerpts and supports commenting and tagging purposes, while Highlight [PKD+12] annotation tool supports classification based on a predefined schema

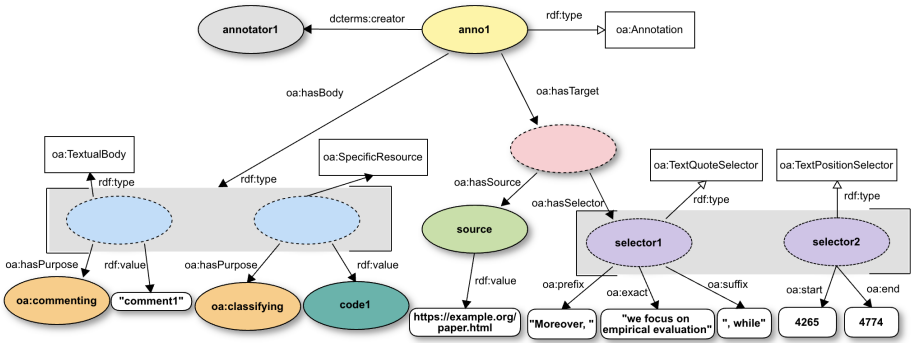


Figure 7.2: An example of a web annotation with more than one purpose (commenting and classifying) and a text excerpt in a html file as Target. Classifying is used to label the annotated target based on a classification schema and comment is used to provide complementary information about the classification decision (e.g., to memo).

(see Fig. 7.3). Here variations bloom. In the same way, as W3C recommendations can be reused, annotation clients' functionality can be also reused. In the next section, we will introduce what type of variations are supported by our solution, *WACline*.

7.3 Software description

SPLs are a well-known software engineering method to create a collection of similar software systems [ABKS13]. The main benefits addressed by SPLs are the time to market, cost, product quality, product line scalability, and productivity compared to traditional software development. We require a platform to support a collection of similar software (i.e., annotation clients), so we resort to SPLs.

WACline is an SPL that aims to help researchers and developers create custom browser extensions for Web Annotation. Specifically, browser extensions serve as the front-end (also known as annotation client) to collect and display annotations hosted in an annotation back-end (also known as annotation server) (e.g., Hypothes.is). *WACline* provides features that can be combined in different ways to create customized annotation extensions to perform annotation practices. These features are not offered in a single product. Rather, domain experts choose those features that account for the annotation practice at hand during development. To this end, *pure::variants* [Beu19] is used as the variability management system.

WACline drives the development of the annotation tool. Following SPL product derivation steps and *pure::variants*' facilities, the creation of the annotation tool entails (1) the **configuration** selecting *WACline*'s features to

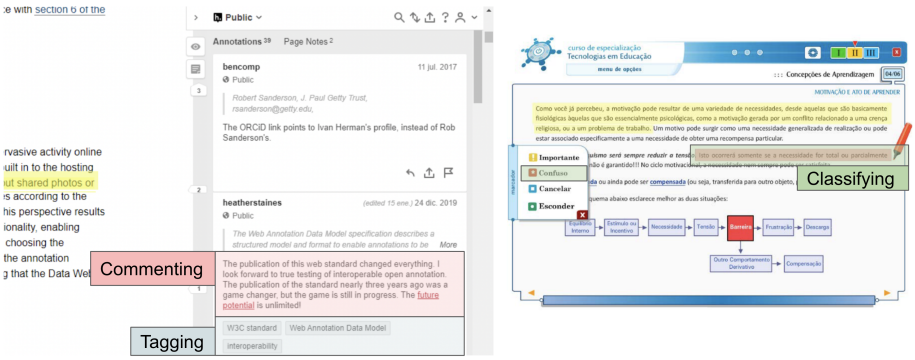


Figure 7.3: Different annotation purposes require different kinds of user interfaces to interact with. A) is an example of *Hypothes.is* where commenting and tagging purposes are supported. B) shows an example of *Highlight* a brazilian annotation tool that supports annotation based on 4 predefined classification values: *importante*, *confuso*, *cancelar* and *esconder*.

meet annotation requirements, (2) the **generation** of the configured product instance, (3) the executable **building** and (4) **testing and delivering**. Although these steps are explained in the user manual[1], next, we briefly describe each of them:

- **Configuration**. The researcher has to choose the set of features that can be reused from *WACline* to meet the requirements of the annotation practice. It is possible to select any feature contained in *WACline*'s feature model (see Section 7.3.2), as long as the dependencies among features are respected (e.g., a feature to filter annotations by users requires a shared remote annotation backend).

- **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 Fig. 7.5). These preprocessor directives are used to discern commonalities (i.e., features that all Web Annotation clients share) from variability (i.e., code implementing specific features functionality). Through a preprocessing stage, a dedicated annotation tool is generated only with the desired features, filtering out the rest of the code.

- **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 extension for the target browser (e.g., *Google Chrome*).

[1]User manual: github.com/onekin/WacLine/blob/master/README.md

- **Test and Delivery**. The created output 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., publishing it in Chrome Web Store).

7.3.1 Architecture

WACline generated tools follow a browser extensions architecture [2] where the *manifest.json* is the entry point for the extension. This file specifies the scripts involved in the extension, their components (i.e., content script, or background), permissions, etc.

In our case, most of the functionality is executed in the Content Script, which augments webpages to support 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*, and so on.

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 [3], Webpack [4], Babel [5] and Gulp [6] 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 [7].

7.3.2 Functionalities

An SPL documents functionality through the Feature Model [LKL02]. It is a representation of the set of features supported and shows graphically the relations among them. *WACline* is the result of knowledge accumulation of three annotation extensions implemented during this thesis plus an extensive review conducted of a selected list of 22 annotation tools in the market in 6 different domains (biomedical, educational, history, journalism, linguistics, peer-reviewing and general-purpose tools) and 4 general-purpose annotation tools [8].

Fig. 7.4 shows the Feature Model for *WACline*, which is divided into six clusters that represent the main components of a Web Annotation client:

- *Annotation Server* gathers features that concern annotation storage (e.g., *Hypothes.is* [Hyp19]).

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

- *Target* clusters features that refer to the annotated content resource (e.g., format, source, or selector).

- *Purpose* refers to what an annotation is created for: classifying, commenting, replying, and assessing. New purposes can be defined following W3C recommendation [SCY17].

- *Operation* 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* 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.

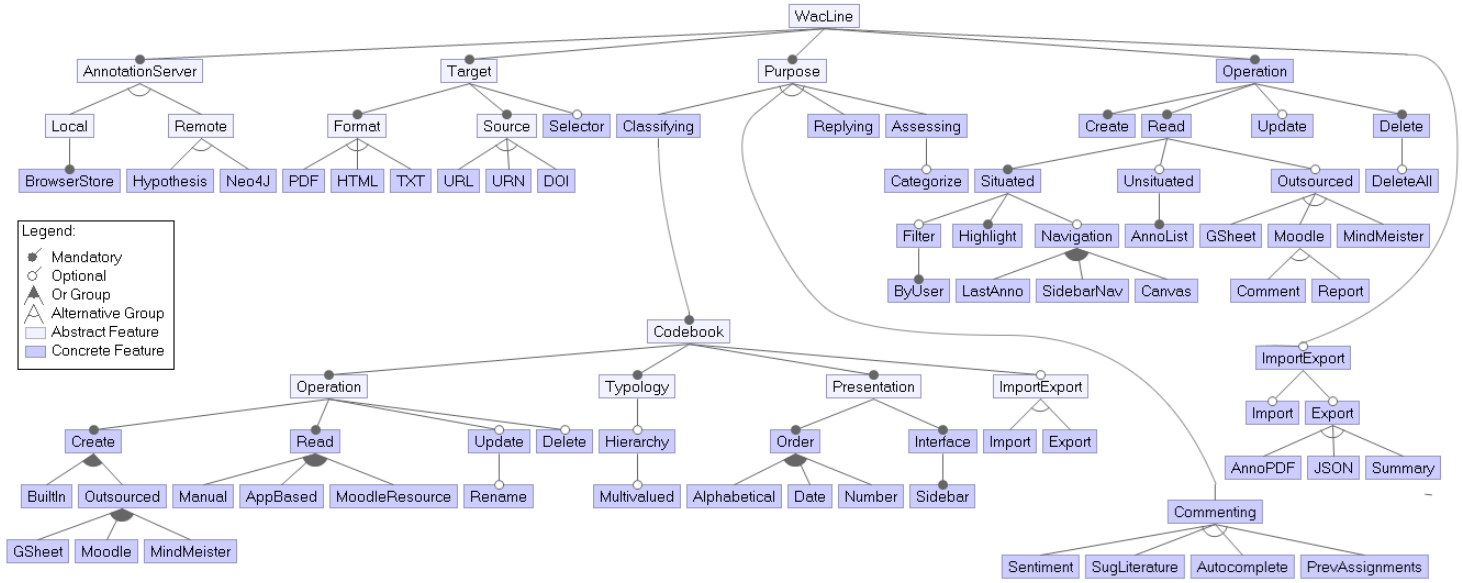
- *Import & Export* permit annotations to be imported and exported in different formats for further reuse outside the extension.

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 their code, two scenarios may emerge:

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

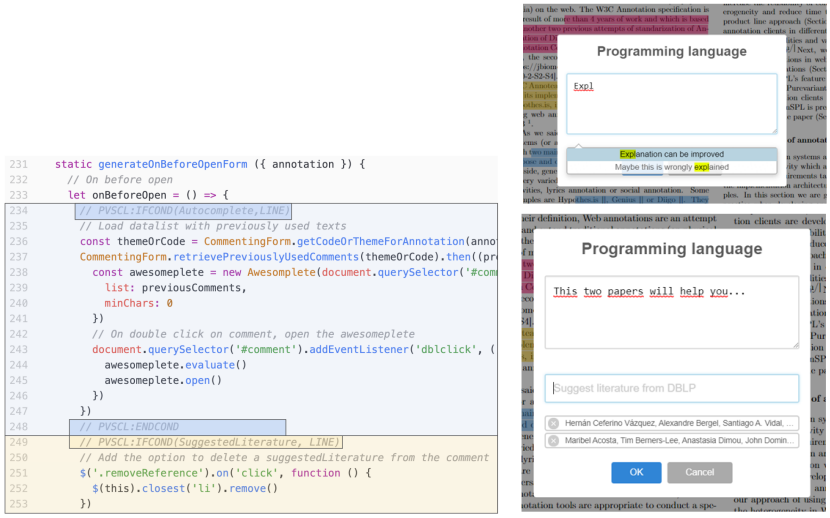
- extending with new features, which involves the integration of brand-new source code that can interact with other existing components of *WACline* via events (see Section 7.3.1).

As mentioned before, *WACline* is an annotated SPL. Annotated SPLs resort to preprocessor directives (also known as *#ifdefs*) to realize variability in code and extensions done over *WACline* (specialized features and implemented new features) should be annotated using *pure::variants* preprocessor directives prior to merge. Fig. 7.5 shows how to annotate code that implement *Autocomplete* and *SuggestedLiterature* features and how the UI and functionality of the annotation tool change depending on the selected configuration. 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 and the user will see the top commenting form variant. If *SuggestedLiterature* feature is selected, it will be displayed as shown in the bottom one in Fig. 7.5(b). While, in this case, both can be selected, providing a UI with both features, enhancing comment by *Autocomplete*, and *SuggestedLiterature* facilities.



Filter.ByUser → Remote
Canvas → Hierarchy
SidebarNav → Sidebar
Replying → Commenting

Figure 7.4: Partial view of *WACline*'s Feature Diagram: structure (top) and a sample of dependencies between features (bottom).



(a) (b)

Figure 7.5: *#ifdef* blocks for the *Autocomplete* and *SuggestedLiterature* features (partial view): source code (left) and GUI variations (right).

7.4 Evaluation

To evaluate *WACline* as a purposeful platform to develop customized web annotation clients, we have conducted a technical feasibility evaluation (i.e., to know whether the SPL has enough variability to develop other types of annotation clients apart from the previously presented examples) and quantification of gains of using *WACline* (i.e., measurement of costs in development and maintenance, and reusability of the source code).

7.4.1 Third-party examples

To evaluate to what extent *WACline* facilitates the creation of other types of annotation tools, next, we present three annotation tools for three different contexts developed by other developers. For each of the cases, it is presented the annotation context, what has been developed or adapted from *WACline*, and the development cost.

Concept&Go

Concept&Go annotation tool has been developed from *WACline* [Gar20]. It is framed as a master's thesis project developed by a computer science student.

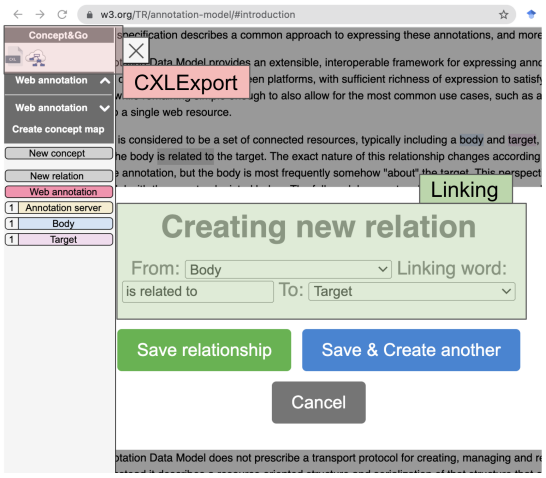


Figure 7.6: *Concept&Go* annotation tool. *Linking* feature creates a button in the sidebar to allow users to relate two concepts using a linking word, in the example “Body" and “Target" concepts are linked using “is related to" linking word. Buttons to export a Concept Map in *CmapCloud* using CXL format are at the top of the sidebar.

Concept Mapping (CM)[9] 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:

- (T1) annotate concept maps' concepts and relationships from different text resources and,
- (T2) 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. 7.6). Firstly, to capture relationships (requirement related to T1), it was needed to adopt *Codebook-based classification* and extend *WACline*'s purposes with a new purpose (*Linking*) to create links between two concepts. Secondly, in order to visualize the map (requirement related to T2), *WACline* was extended with *CXLExport* feature to export the gathered concepts and relationships with the annotations to CmapTools cloud [10].

[9] 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 to describe their structure of knowledge, and they promote knowledge sharing and the creation of new knowledge from them [WM09].

[10] https://cmapcloud.ihmc.us/

Concept&Go was benefited by 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 a total of 3160 SLOCs. The time-to-market of *Concept&Go*, including the SPL understanding, adoption, and extension of these features took around three months by a single graduate student, where most of the invested time was invested in concept map creation functionality.

Docal

Docal annotation tool was developed from *WACline* [DdOH20]. It is framed in a bachelor's degree thesis developed by a computer science student. *Docal*'s main goal is to facilitate law researchers to review law case documents to determine the legal problem and process the available legal information related to the problem they have identified [LAFM16]. Currently, researchers conduct this process annotating manually in paper format. To improve and facilitate the gathering, processing, and organization of this data, web annotation has been proposed in this work. The common practice in multiple case law analysis is compound by the following tasks:

- (T1) Retrieve a list of law cases about a specific matter (e.g., "legislation and measures in the pandemic season") from databases such as Thomson Reuters Aranzadi.
- (T2) After registering them in a new analysis session it is necessary to conduct a grounded theory to emerge different topics [GH13a]. In this process, it is common to relate (link) evidence and concepts that can be in the same case law or maybe in a different one. Here, a customization of *Linking* feature developed in *Concept&Go* has been done (see Fig. 7.7), supporting not only the relation of concepts in the same document but also the annotations done in different ones.
- (T3) After analyzing different legal documents, jurists and researchers have to summarize in a document all findings to later determine possible laws to apply to the fact under study. To help in this process, *Docal* implements a new feature to export analysis results to a head-start word document gathering all annotated findings.

Docal benefited from 24 *WACLine* features (up to 12540 SLOC). Only three new features were necessary to support this new annotation practice (*Session*, *Linking* and *DocExport*), though they account for 1365, 1065 and 401 SLOCs respectively, making a total of 2831 new SLOCs implemented. The time-to-market of *Docal*, including the SPL understanding, adoption, and extension of these features took around four months by a single undergraduate student.

Fival

Fival annotation tool was developed from *WACline*. It is framed as a bachelor's degree thesis by a computer science student. *Fival*'s main goal is to facilitate

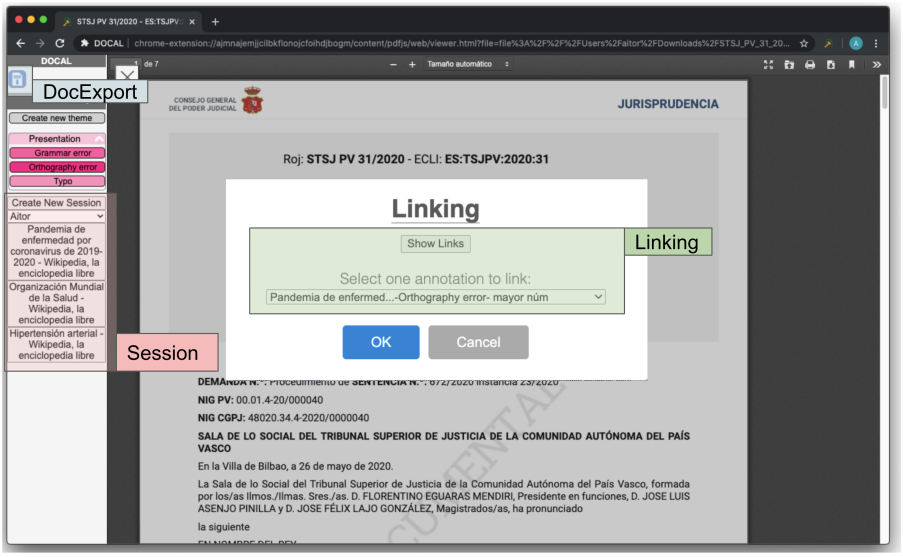


Figure 7.7: *Docal* annotation tool showing DocExport button to create a report, session management in the bottom part of the sidebar and linking UI where are displayed existing links for selected annotation and a dropdown list with other linkable annotations.

the bachelor's degree thesis examination committee to review and evaluate the documentation provided by the student. To this end, a list of tasks should be conducted by lecturers:

- (T1) A group of lecturers (i.e., the examination committee) has to read an extensive document (usually in PDF) provided by the student.
- (T2) They have to take notes and evaluate following common evaluation criteria defined by the department, degree, or committee itself. Some of those notes can be personal (for own consumption by the lecturer to decide the mark or to help to formulate questions on the day of the defense), while others should be shared with the rest of the examination committee to decide a final mark for the student. To this end, the *Grade* feature has been implemented (see Fig. 7.8). It reuses *CodebookImport* but is extended with gradable items. The lecturer imports a codebook where items can be weighted to automatically calculate a final mark for the student. Additionally, the tool creates a text input where a numeric mark can be provided for each of the items in the codebook. Finally, it automatically calculates the current mark for the student
- (T3) All the lecturers, based on shared notes, have to reach an agreement to decide on a final mark for the student and write a final report justifying the positive and negative aspects of the work done by the student. To this

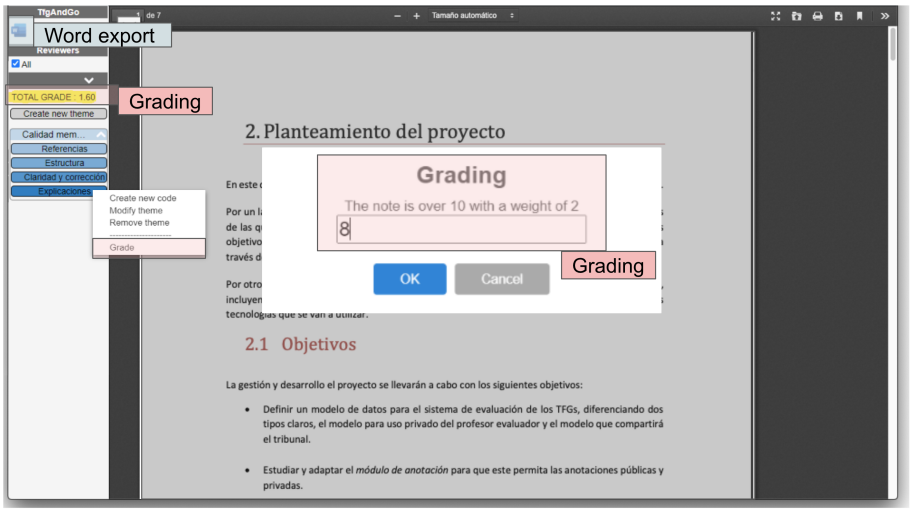


Figure 7.8: *Fival* annotation tool's main interface where *Grading* and *WordExport* features are shown. Grading menu is shown by right-clicking a theme or code in the codebook sidebar and allows lecturers to give a number mark to the selected rubric item. *WordExport* is shown on top of the left sidebar.

end, *AnnotationScope* feature has been implemented. It allows lecturers to set whether an annotation is for private consumption or will be shared with the rest of the lecturers' board

- (T4) Evaluation rubrics and reports can be different depending on the department, degree, or university where the evaluation is conducted. To this end, a new exporting feature has been implemented: *WordReport*. It generates a word report based on a word template and is fulfilled by annotated content and marks for each of the rubric criteria.

Fival was benefited by 19 *WACLine* features (up to 10345 SLOC). Only three new features were necessary to support this new annotation practice (*Grading*, *WordExport* and *PublicPrivate*), though they account for 316, 156, and 176 SLOCs respectively, making a total of 648 new SLOCs implemented. The time-to-market of *Docal*, including the SPL understanding, adoption, and extension of these features took around three months for a single undergraduate student.

7.4.2 Gains in development and maintenance

In this section, we quantify to what extent the source code among three main products has been reused.

Gains in development: *WACline* follows a reactive approach to SPL development as we have shown in Chapter 6. First, an up-front investment was made to build up the core based on refactoring *Highlight&Go*, *Mark&Go* and

Review&Go. *WACline* accounts for a total of 16,179 SLOC, where the core has 9,700 Source Lines of Code (SLOC) and variability (i.e., annotated code using *#ifdef* clauses) account for 6,479. The core was eventually confronted with different annotation practices: literature reviews, assessment marking, and peer review. As a result, three annotation tools were developed that accounted for each of these practices' specifics: *Highlight&Go* (2,936 additional LOC), *Mark&Go* (2,949 additional LOC) and *Review&Go* (594 additional LOC) at mid 2021 (release v0.3 of *WACline*). Further changes have been conducted to solve some minor bugs and improvements in all three annotation tools that could change a minor proportion of some of the presented results. Apart from the core, which accounts for 59.95% of the total number of lines of *WACline*, the three products share some of the implemented features. Fig. 7.9 shows the variability part that they share (i.e., the features that they share) in a Venn diagram:

- All three products share a 57.62% of SLOCs implementing variability, where features implemented in *Highlight&Go* and non-used by any other annotation tool accounts for 6.57% of SLOCs (425 SLOCs)
- *Mark&Go* required the development of 17.29% of SLOCs and it reuses 1.49% of SLOC with previously implemented *Highlight&Go* features.
- *Review&Go* has reused one feature from *Mark&Go* and up to 6 from *Highlight&Go*, requiring only the implementation of 3.53% of SLOCs

Gains in maintenance. One of the benefits apart from the development cost in SPLs is the cost of maintaining and evolving applications. To the day of this writing, we have solved more than 57 issues addressing a bug [11]. The Table 7.1 shows the number of bugs solved per year, the bugs affecting to only one of the products or more than one and the mean of the number of LOCs required to fix one bug [12]. The results show that since the creation of the second product (*Mark&Go*) in 2018 there were 29 bugs affecting only one product (accounting for a total of 930 LOCs), 5 affecting two products (accounting for a total of 196 LOCs), and where 23 of them affected to the core or a feature shared by all three (accounting for a total of 2885 LOCs). The bottom line is that as more bugs affect more than one product, and consequently, more source code has to be fixed, the more benefit would be reached by an SPL approach as it does not require fixing the same bug in different products.

7.4.3 Threats to validity

This work advocates for SPLs to performantly cope with annotation variability. The conjecture is that "performant heterogeneity accomplishment" (dependent variable) can be achieved by using a given software engineering method (independent variable), i.e., SPLs (in contrast with, e.g., *clone&own*). We address this hypothesis through three case studies (i.e., *Highlight&Go*, *Mark&Go*, and

[11] https://github.com/onekin/WacLine/issues

[12] The complete analysis is available here: https://rebrand.ly/waclineMaintenance

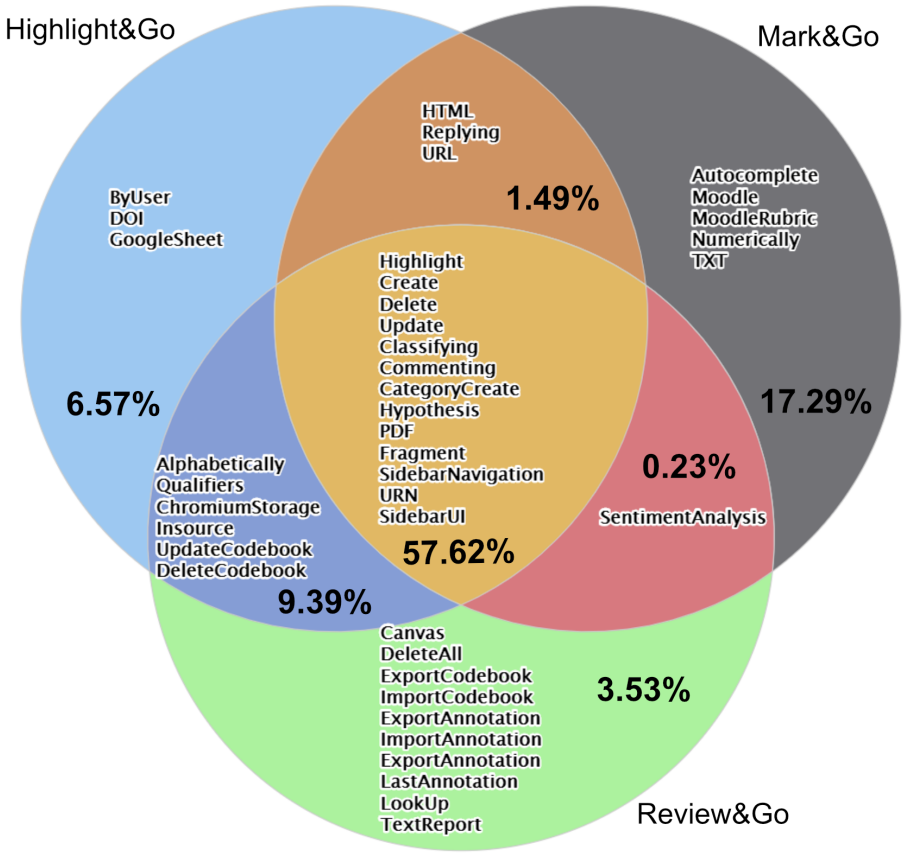


Figure 7.9: *WACline* reusability map in terms of LOCs. Reusability rates are 57.62% for three, 11.11% for two, 27.39% for one product, while 3.88% of LOCs are planned to be used by new products (e.g., *Concept&Go*).

Review&Go) for which the effort has been reduced, and for other three case studies (*Concept&Go*, *Docal* and *Fival*) were also analyzed in terms of reuse and time-to-market. The next paragraphs look into the validity of these results.

Construct Validity. We resort to LOC as the operational measure to assess both reduced effort and time-to-market. Although LOC is traditionally considered a proxy for development effort, not all codes are the same. Code with much Conditional Compilation is more difficult to develop [MRG+18].

Internal Validity. This aspect of validity looks into the extent the SPL (independent variable) is the ultimate cause of "performant heterogeneity accomplishment" (dependent variable). From this perspective, *WACline* illustrates SPL benefits (for a literature review on this topic, refer to [BJBCL18]). Yet, it could be argued that a co-founding variable, i.e., the developers themselves, might also impact the result. That is, the fact that *WACline* developers also

Table 7.1: Bug fixing for *WACline*: (1) number of corrective issues solved from 2017 to 2021; (2) number of bugs affecting 1, 2 or 3 products and in parenthesis the number of LOCs modified to solve the issues.

Year	# of bugs	affecting 1 prod. (#LOC)	affecting 2 prod. (#LOC)	affecting 3 prod. (#LOC)
2018	4	2 (119)	2 (65)	0
2019	20	11 (415)	1 (2)	8 (291)
2020	23	10 (292)	1 (125)	12 (2392)
2021	10	6 (104)	1 (4)	3 (202)
Total	**57**	**29 (930)**	**5 (196)**	**23 (2885)**

participate in the derivation of the products (i.e., *Highlight&Go*, *Mark&Go*, and *Review&Go*) makes it natural for them to reuse what they have already developed. This certainly deserves further evaluation. Additionally, third-party-implemented three annotation tools (*Concept&Go*, *Docal*, *Fival*) could work as a first attempt for investigating whether other communities can easily extend *WACline* to account for their annotation specifics. Results in terms of time-to-market are promising but should be proven the suitability of the solution by experienced web annotation developers.

External Validity. This aspect of validity is concerned with the generalization of the findings. *WACline*'s feature model is much influenced by the set of annotation tools being revised and implemented examples in this work. Analyzed third-party annotation tools were selected based on availability and popularity, trying to select at least one for each of the 6 domains analyzed where annotation tools have been used. Yet, other communities might exhibit annotation workflows not considered by *WACline*. That said, *WACline* is a proof-of-concept for the hypothesis about SPL suitability. Hence, annotation communities should exhibit variability in their practices, we might reasonably argue that an SPL approach would also report for them similar benefits that those of *WACline*. Another concern is about the *WACline* infrastructure, specifically annotation servers (i.e., Hypothes.is) and annotation clients architecture (i.e., Chrome Browser Extension). Annotation Servers other than implemented (*Hypothes.is*, *Neo4J*, *LocalStorage*, and *Google Sheets*) might be used. This would require adding appropriate drivers to *WACline*. On the other hand, *WACline* products are fleshed out as Chrome's extensions. Specifically, *WACline* follows Chromium, i.e., an W3C's in-progress recommendation for cross-browser extension portability. This implies that *WACline* tools can run upon any browser supporting Web Extensions API [Moz22a]. At the time of this writing, this includes Chrome, Opera, Brave, Edge, and partially, Firefox [Moz22b]. Unfortunately, Safari, Android-based browsers, and iOS-based browsers are not yet Chromium-compliant. Therefore, browser extensions may not be a valid solution for annotation scenarios that require a tablet or a smartphone.

7.5 Impact

In the last six years, up to seven different surveys were published [NŠ19, GSA18, KMK16, GCSCS18, BMB17, KTV18, CKK21] which analyze more than 200 annotation tools used in linguistics, education, biology, and e-health research. This denotes the high number of research projects that can be potentially benefited from the use of *WACline*.

WACline represents a step toward annotation tooling development for the review process in multiple research disciplines. By adopting and extending *WACline*'s features, researchers and annotation tool developers 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 variability in any of the six annotation tools (i.e., features).

Furthermore, if the creation of the annotation tool has led to the development of new functionalities, researchers can incorporate those novelties to *WACline* in terms of new features. To integrate feature modifications and the implementation of new requirements, the annotation tool developer must annotate the new code with the preprocessor directives and pull them to the repository. These extensions can be shared among the community to create variants that can support new annotation practices, combining some of these new features. This is the case of one of the implemented features in *Concept&Go*, *Linking* purpose, which was adapted to be used in *Docal*.

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

The bottom line is that as development and maintenance costs are reduced, funds for projects where annotation tool development is involved can be better invested, for example, increasing efforts to better evaluate and justify the relevance of their research work.

As mentioned before, W3C recommendation already exists [W3C17] but the majority of annotation tools do not follow any standard to represent annotations' data model [KMK16]. This hinders important aspects of research, such as reproducibility or reuse of annotation data sets, requiring transformation between annotation tools' data models [AMD+19] or integrations, plug-ins, or tools, like the ones made for Hypothes.is [13].

Reproducibility and reuse of annotations have been solved by the W3C annotation data model. However, there is still a gap between W3C recommendations and real practice where even new annotation tools keep using custom formats. *WACline* follows the 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 [CSRC13]. The interoperability problem has been discussed in some sessions

[13] https://web.hypothes.is/tools-plug-ins-and-integrations/

at the IAnnotate conference [14], one of the reference conferences about Web Annotation, and this approach can benefit annotation practitioners in different areas.

7.6 Conclusion

In this chapter, we have presented the resultant SPL to manage heterogeneity in Web Annotation Clients. First, we have described W3C's Annotation model including its main variability aspects (e.g., Target, Purposes). Second, we have introduced the design of *WACline* after conducting a knowledge accumulation process defined in Chapter 6 and based on a review of the annotation tools market. Third, we have introduced how *WACline* has been evaluated in terms of feasibility (i.e., presenting three new annotation clients implemented by third-party developers) and reusability (i.e., quantifying to what extent the implemented source code has been reused across projects during development and maintenance). Finally, we have presented the impact that this SPL can have on the development and maintenance of annotation clients.

Part of this chapter has been published in Elsevier Software X:

- Medina H., Diaz O. & Garmendia X. WACline: A Software Product Line to harness heterogeneity in Web Annotation. SoftwareX (2022) Vol. 18C. **JCR Q3**.

[14] Discussion notes about interoperability at the IAnnotate: https://rebrand.ly/iannotateInteroperabilityDiscussionNotes

Chapter 8

Conclusions

8.1 Overview

This thesis uses ADR as the research methodology, which is based on DSR. Hevner [Hev07] introduced a three-cycle view of DSR in which rigor, design, and relevance intertwine over the research framework which includes the knowledge base, environment, and research project. The rigor cycle connects the design science activities with the knowledge base, while the relevance cycle bridges design science activities with the environment (see Fig. 1.6). This chapter summarizes the main results contributed to the knowledge base and the environment. First, we introduce the main results to the knowledge base in Section 8.2 and resultant publications in Section 8.3. Second, ADR stresses the relevance cycle, which has a strong influence on the environment (social and technological). Section 8.4 shows the practical impact that this thesis has on the environment. Third, we left the doors open to further improvements in the environment and knowledge base in Section 8.5. Finally, research means not only contributing to the environment and knowledge base but also sharing knowledge and enhancing the environment outside of your organization. I had the opportunity to work in a research stay during my Ph.D., which is described in Section 8.6.

8.2 Results

This thesis proposes the use of SPLE to create customized web annotation tools to address efficiency and effectiveness problems in different review practices. To this end, we have defined the main research question and three sub-research questions for each of the contexts we have focused on. We should begin by recalling the thesis's main research question:

> How to design a platform to systematically reuse features (ARTIFACT)
>
> that satisfies heterogeneity and extensibility (REQUIREMENT)
>
> so that developers reduce the development and maintenance cost (STAKEHOLDER GOAL)
>
> in the creation of web annotation extensions for reviewing? (CONTEXT)

To address this RQ, we propose the use of SPLs to create customized web annotation tools in the review domain. We constructed an SPL to face heterogeneity in annotation practices using a reactive approach described in Chapter 6. The resultant SPL is called *WACline*, described in Chapter 7. We have evaluated the SPL in terms of feasibility to create new annotation clients and reusability quantifying the source code has been reused across projects.

As mentioned before, in this thesis, we have addressed three sub-research questions for literature reviewing data extraction, student assessment, and scholarly peer review.

In the context of **literature reviewing**, we have addressed the following research question:

> How to design a dedicated annotation tool
>
> that satisfies portability
>
> so that researchers conduct data extraction effectively and efficiently
>
> in secondary studies' data extraction process?

We have addressed this RQ in Chapter 3. We first analyze current practice and tool support to look for problems that arise in this context and later propose the use of a customized web annotation tool to conduct data extraction. We define requirements for the annotation tool and provide an instantiation with *Highlight&Go* that accounts for efficiency (i.e., automating the translation of classification decisions) and effectiveness (i.e., enhancing spreadsheets to increase consistency, traceability of taken decisions by allowing moving back to the paper's evidence, and completeness). An evaluation was conducted in a real setting with Ph.D. students that revealed positive results in individual data extraction efficiency and effectiveness.

In the context of **students' assessment**, we addressed the following research question:

> How to design a dedicated annotation tool
>
> that satisfies seamless integration with LMSs
>
> so that lecturers can increase the feedback quality
>
> in higher education at scale?

We have addressed this RQ in Chapter 4. Based on current practice and informed by theories of quality feedback and cognitive behavior therapy, we propose the use of customized web annotation tools to account for timely, specific, contextualized, and personal feedback. We define the requirements for such a tool and provide an instantiation in *Mark&Go*, an assignment marking tool integrated with Moodle. An evaluation in a real setting reveals positive results and possible improvements to generalize its use.

In the context of **peer review**, we addressed the following research question:

> How to design a dedicated annotation tool
>
> that provides guidance
>
> so that reviewers can increase the feedback quality
>
> in scholarly peer review?

We have addressed this RQ in Chapter 5. To answer this RQ, instead of resorting to general-purpose tools like Acrobat Reader, we propose the use of customized web annotation tools that account for review specifics. We define the requirements for such a tool and provide an exemplary instantiation in *Review&Go*. A preliminary evaluation reveals positive results in terms of perceived usefulness and ease of use.

8.3 Publications

Part of the work presented in this thesis has already been presented and discussed in different peer-reviewed forums. The publications that endorse this thesis are listed below.

Selected publications

- Díaz, O., Medina, H., & Anfurrutia, F. I. (2019). Coding-Data Portability in Systematic Literature Reviews: a W3C's Open Annotation Approach. Proceedings of the Evaluation and Assessment on Software Engineering, EASE 2019, Copenhagen, Denmark, April 15-17, 2019. ACM. **CORE A, Class 3**. Related to Chapter 3.
- Díaz, O., Contell, J. P., & Medina, H. (2019). Performant Peer Review for Design Science Manuscripts: A Pilot Study on Dedicated Highlighters. Extending the Boundaries of Design Science Theory and Practice - 14th International Conference on Design Science Research in Information Systems and Technology, DESRIST 2019, Worcester, MA, USA, June 4-6, 2019. Springer. **CORE A, Class 3**. Related to Chapter 5.
- Diaz, O., Medina, H., & Perez Contell, J. (2021). Promoting Design Knowledge Accumulation Through Systematic Reuse: The Case for Product Line Engineering. 54th Hawaii International Conference on System Sciences, HICSS 2021, Kauai, Hawaii, USA, January 5, 2021. ScholarSpace. **CORE A, Class 1, Nominated for best paper award**. Related to Chapter 6.

- Medina H., Diaz O. & Garmendia X. WACline: A Software Product Line to harness heterogeneity in Web Annotation. SoftwareX (2022) Vol. 18C. **JCR Q3**. Related to Chapter 7.

Publications and presentations in International Conferences & Workshops

- Medina, H., Díaz, O., & Anfurrutia, F. I. (2018). Highlight&Go: una extensión para automatizar la extracción de datos en revisiones sistemáticas de la literatura utilizando Google Sheets. Actas de las 23. Jornadas de Ingeniería Del Software y Bases de Datos, JISBD 2018.

- Medina, H., Diaz, O., Contell, J. P. (2018). The Cold-star Challenge: Introducing annotation at the University of the Basque Country. IAnnotate'18, the 6th annual conference for interoperable annotation technologies and practices.

- Medina, H., Diaz, O. (2019). Web Annotations for Assignment Marking: Challenges and opportunities. IAnnotate'19, the 7th annual conference for interoperable annotation technologies and practices.

- Diaz, O., Contell, J. P., & Medina, H. (2020). Software scaffolds for quality feedback in peer review. Peere'20 The International Conference on Peer Review, 1–4.

Publications under review

- Diaz O., Medina H., Azanza M. (submitted for review to Elsevier Computers&Education in 2022). Balancing Quality and Timeliness in Student Feedback at Scale: A Case of Action Design Research. Related to Chapter 4.

- Medina, H., Azpeitia, I., Anfurrutia, F. I., Díaz, O. (submitted for review in 2022 to Elsevier Information and Software Technologies). Supporting efficient and effective data extraction through annotation tooling. Related to Chapter 3.

8.4 Practical impact

In this thesis, the conducted research not only influenced the knowledge base but also the application context or environment [HPCWA18]. Gill argues that rigor and relevance should be communicated in an effective way to practitioners [GH13b]. In this thesis, we attempted to go beyond contributing to the knowledge base by attempting to resonate among practitioners. To this end, we facilitate the use of our annotation tools to real practitioners by making them available through the Chrome Web Store and providing explanatory user manuals and guidance videos. In the same way, we facilitate the use of *WACline* by

providing manuals for contributors and annotation client developers[1], as well as videos on how to configure a customized annotation client. Documenting the use of software artifacts is considered an important factor for communicating research outcomes to end-users, especially in an ADR setting where real practitioners are involved in the development, evaluation, and further use, where utility is realized [MH19]. We have looked in each of the chapters at the users across the three main outcomes *Highlight&Go*, *Mark&Go*, and *Review&Go*. The presented graphs, taken from the Chrome Web Store, showed that *Highlight&Go*'s accounts between 50 and 70 users in the last two years, *Mark&Go*'s between 40 and 60 users, and *Review&Go*'s around 20. Some of them are users from our research group or department, but looking at how they are geographically distributed, nearly one-third of the users for the three tools are from America (specifically from the USA, the venue for IAnnotate conferences) or Asia, which is quite surprising.

WACline is an alive and thriving ecosystem of web annotation clients that goes beyond what is explained in this manuscript. It also has served as the basis for several Bachelor's degrees and Master of Science degree final projects. This author has also been involved in up to 6 bachelor's and master's theses, and of course, the created ecosystem around *WACline* and its tools would not be possible without the participation of these students. Moreover, the knowledge they acquired during these final projects would not be possible without this thesis. Next, we will provide a short description of these projects:

- *Perez, E. (2018). Visualización de anotaciones web para la calificación de exámenes.* *http://hdl.handle.net/10810/29101*. This BSc thesis aims to provide visualizations over annotations created using *Mark&Go*. These visualizations would facilitate lecturers to analyze the assessment results, giving an overview of how students perform during the course.

- *Garmendia, X. (2019). Aplicación de una arquitectura de líneas de producto para una familia de anotadores Web.* *http://hdl.handle.net/10810/36027*. This BSc thesis was the initial attempt at the design and development of *WACline* based on the annotation tools that existed at the time: *Highlight&Go*, *Mark&Go* and *Review&Go*.

- *Díaz de Otazu, A. (2020). Desarrollo de una aplicación para el análisis de sentencias judiciales utilizando la línea de productos software WacLine.* *http://hdl.handle.net/10810/48781*. In this BSc thesis, *Docal* was developed, the annotation tool for case law document analysis created from *WACline* that is presented in Section 7.4.1.

- *Arce, G. (2020) Desarrollo de una aplicación para apoyo en la evaluación de TFGs utilizando la línea de productos software WacLine.*. In this BSc

[1] Manuals for annotation client developers include instructions for configuring, building, testing and contributing to *WACline*. As part of this documentation, we provide a conceptual model, a feature model documentation with a description of each of the existing features, and architecture diagrams that are hosted at https://github.com/onekin/WacLine

thesis, *Fival* was developed, an annotation tool for final degrees evaluation presented in Section 7.4.1

- *Garmendia, X. (2020). Feature-based software development: a case for Web annotation-based tools.* In this MSc thesis Concept&Go was developed, an annotation tool for creating concept maps presented in Section 7.4.1.

- *Bereciartua, I. (2021). Una herramienta para la revisión de manuscritos de investigación: un enfoque basado en Líneas de Producto. http://hdl.handle.net/10810/53318*. In this BSc thesis the functionality to support Empirical Standards and keyword lookup in *Review&Go* presented in Section 5.5.1 was developed.

8.5 Future work

The Ph.D. is just the beginning of the research journey. This thesis is not an exception and opens several issues, probably more than those addressed during this journey. The next section discusses the limitations of each piece of work, as well as some of the questions that this thesis leaves open in the addressed study fields.

8.5.1 SLR support using web annotation and *Highlight&Go*

- As mentioned in Chapter 3 a more extensive evaluation should be performed. This evaluation could be targeted at more experienced researchers, but it could also compare the performance of conducting data extraction between *Highlight&Go* and other data extraction tools, such as QDA tools (e.g., nVivo).

- SLR data extraction is a single step in the long journey of conducting a literature review. From the results of the *Highlight&Go* project, it can be interesting to analyze to what extent *Highlight&Go* (or a variation of *Highlight&Go*) is suitable to support other steps such as selection of studies or piloting.

- Web Annotations in a spreadsheet are the main outcome of the data extraction where every highlight is registered. This makes it possible to audit data extraction and reuse all the processes by third-party researchers (e.g., to conduct an SLR update). However, further investigation is needed to validate this.

- We have evaluated briefly with practitioners to what extent the resultant spreadsheet is useful and can be combined with other tools for analysis and reporting. Some suggestions from participants were to integrate *Highlight&Go* with mind-mapping tools for thematic analysis, or better integration with quantitative analysis tools like SPSS or R.

8.5.2 Support for assessment in education using web annotation and *Mark&Go*

- As mentioned in Chapter 4 to analyze to what extent our results are generalizable, an evaluation that goes beyond the scope of our department could yield interesting insights. The evaluation episodes prove, based on the qualitative opinion from the lecturers, that *Mark&Go* improves the feedback quality. However, the evaluation context is a Computer Science degree or assignments related to Computer fundamentals, where especially textual reports and source code have been assessed, and lecturers are keen on technology. Evaluation in other educational disciplines, where lecturers are likely to be more unfamiliar with software applications, could show whether *Mark&Go* increases the quality of the feedback.

- In the same way, further evaluation is needed to prove the quality of feedback reached by *Mark&Go*. Initial tests have been done to encourage students to install *Mark&Go* to access their feedback. However, we realized that they are quite reluctant to install the extension just to simply access their assessed assignment. To facilitate the accessibility of the feedback, we are working on providing the feedback (notes, colored highlights, etc.) in a single file that can be opened.

- Based on the feedback provided by *Mark&Go*, and related to what is implemented in one of the Bachelor's degrees in the ecosystem of this thesis, visualizations and reports over *Mark&Go*'s annotations can be done (e.g., a final report including the most common mistakes). This presumably will facilitate lecturers in identifying gaps in the learning process, teaching materials, or tuning assignment difficulties.

- Thanks to the W3C Web Annotation standardization, annotations are digitally created and can be consumed to learn from them and aid in the assessment process (e.g., automatically spotting mistakes, expressions, or phrases in students' assignments based on previous corrections with the tool).

8.5.3 Supporting peer-reviewing using web annotation and *Review&Go*

- As mentioned in Chapter 5 an evaluation was performed in a testing session with nine participants. Evaluation episodes are far from proving *Review&Go*'s effectiveness in achieving its objective. As a means to comprehensively test the tool, an experiment can be conducted comparing review performance and review quality with and without using *Review&Go*.

- Even if *Review&Go* has been evaluated by researchers with a large trajectory in research and peer-reviewing, only the reviewer stakeholder has been taken into account. In peer-reviewing, up to four stakeholders (authors, reviewers, editors, and readers) are interested in providing the best

(i.e., most effective) and fastest (i.e., most efficient) reviews possible. With *Review&Go* we use web annotations that later can be consumed by meta-reviewers and journal editors to facilitate decision making. Combining a common review framework (e.g., ACM Empirical Standards) and annotations from multiple reviewers can facilitate decision-making in meta-review activity. In the same way, the results of those reviews and meta-reviews are consumed by the authors who are responsible for improving their papers. To this end, annotations could also help in spotting the main changes to be addressed and in the comprehension of reviewers' comments as they are provided in the context. Additionally, some journals are starting to make the reviews public to end readers, which is called Open Peer Review [WRAW00]. Open peer review and web annotation can be combined to enhance the published paper with annotated content in the previous phases.

8.5.4 WACline

- *WACline* has been tested as a feasible platform to create custom web annotations in up to 6 different reviewing contexts. However, a study of the replicability of other annotation tools can be conducted to validate to what extent it is a feasible solution to create other types of annotation tools. We have identified nearly 200 annotation tools that presumably can be replicated or created from *WACline*[2] to validate its extensibility.

- *WACline* heterogeneity is mainly focused on *how* users annotate (i.e., the Body), but we addressed in less proportion *what* users can annotate (i.e., the Target). Currently, only text annotation is supported in multiple formats (PDF, TXT, and HTML) and hosted at different places (Moodle, Digital Libraries, Locally), but not other formats or types of fragments that can be revised. The W3C Annotation recommendation supports the representation of fragments of multiple types of files that would require other user interactions and mechanisms. Currently, there already exist annotation tools that support the annotation of images, videos, 3D models, and so on. This is the case with tools like Recogito, which supports the review of ancient manuscripts and maps formatted as images [SBID17].

- *WACline* is an open-source medium-sized SPL (with around 110 features). In the area of SPL, there is a lack of real examples available for research, but also teaching. *WACline* can also be evaluated as a valid example in SPL courses. It is richer than samples provided by currently existing frameworks (e.g., pure::variants), but still more simple than an industrial SPL, which in most cases their access is limited as they are not open source.

- Accessing real Software Product Lines from companies is difficult, as they are one of the most important assets of the company. This hinders research

[2] https://rebrand.ly/annoToolsList

in the Software Product Lines area. *WACline*, as it is an open-source SPL, can be used as a running example to test and validate research hypotheses in this area. Currently, it has been used in two already-published papers [AIMD21, MD22].

8.5.5 Third-party created annotation tools evaluation

- Taking the *WACline* platform as the basis, currently, three additional web annotation tools have been created as part of different bachelor's and master's degree thesis: *Docal*, *Fival* and *Concept&Go*. They are web annotation tools created from *WACline* that try to solve different problems in concept mapping, assessment of final degree theses, and analysis of case law documents. For the design of these tools, an initial design was planned with practitioners in those areas, but an evaluation is still pending with real practitioners. The next step in these projects should go on evaluating and publishing the results of using web annotation to validate to what extent these tools facilitate these practices.

8.6 Research stage

As we have mentioned before, research means improving the knowledge base and the environment. Knowledge base and environment can be improved in your organization, but research is a "give-and-take" process, where knowledge and improvements should be shared with other communities. In the same way, as we have published our results in international forums, during the course of this Ph.D. work, the Ph.D. has also given me the chance to work and share ideas with researchers in a foreign environment. I did a four-month research stage under the supervision of Professor Claudia Müller-Birn, from the Human-Centered Computing at Freie Universität in Berlin (Germany). This visit, despite the arrival of the COVID-19 pandemic (and its consequent isolation) at the very beginning of my stay, was quite useful to validate our ideas with real web annotation tools developers. In the same way, and thanks to their expertise in the area of Human-Computer Interaction, we have improved the usability of tools that we created from *WACline*, something that made a difference in the practical impact of the artifacts developed in this thesis.

8.7 Conclusion

Annotations are notes at the margin used for centuries. With the advent of digitization, web annotation popularity has increased in the latest years to support multiple activities in areas like education, research, biomedical, and history, just to name a few. From the very first web annotation tool, hundreds of web annotation tools have been developed. To facilitate interoperability and representation of data (i.e., web annotations), the W3C released recommendations

for web annotations. However, it leaves unconstrained the design of web annotation systems' user interfaces. As one annotation tool does not fit all annotation practices, customization of web annotation is required, but usually, this is tackled by re-implementing new annotation tools from scratch, and consequently reinventing the wheel once and again.

We have presented three contexts where web annotations convey as a mediator to solve efficiency and effectiveness problems in the contexts of literature review, assessment in higher education, and peer review. From these use cases, we have designed and developed an SPL to facilitate the reuse of source code avoiding reinventing the wheel. The bottom line is that SPLs can help improve code reuse across annotation projects, hence reducing development costs.

Appendix A

Highlight&Go's confirmatory focus group guide

This section presents the focus group interview guide of *Highlight&Go*[1] used to facilitate the instructor labor to moderate, posing questions and adding prompting follow-up questions to encourage the discussion between practitioners[2]:

- **Introduction** [10 min]
  - Welcome and ask for permission to record
  - Introduction to the purpose of the focus group
- **General questions** [20 min]
  - Ask participants a short introduction about their research experience and context, where *Highlight&Go* has been used
  - Workflow: Which one was your data extraction workflow? What was the goal of the selected strategy?
  - Guidelines: Which guidelines have you followed to conduct your secondary study? To what extent *Highlight&Go* is suitable for the data extraction process you followed?
  - Integration: What other tools were used besides *Highlight&Go*?
  - Limitations: Were you able to use *Highlight&Go* to fully conduct data extraction? Did you need the help of a technician/expert in the tool? Which functionalities did you expect that you did not find in *Highlight&Go*?

[1] The full version can be found in: https://rebrand.ly/highlightAndGoCompleteFocusGroupGuide

[2] Additionally we have shared a short version of the guide with practitioners before the meeting: https://rebrand.ly/highlightAndGoSharedFocusGroupGuide

  - Satisfaction and intention to use: What is your overall satisfaction with the tool? Are you planning to use it the next time you have to conduct data extraction in a literature review?

- **Mechanism utility** [10 min]: we questioned participants to rank *Highlight&Go*'s mechanisms of utility for efficient and effective data extraction
  - Transparent extraction of metadata
  - Transparent storage of coding event logs in the spreadsheet
  - Codebook-based color-coding primary study annotation
  - Overview and detail interface (highlighter with number of annotations + navigation through evidence + canvas like view)
  - Zooming interface on the spreadsheet
  - Hyperlinks to coding context
  - Theme's and Primary studies font-color semaphore and warnings
- **Stage 0: Codebook definition** [5 min]
  - To what extent have you modified your data extraction form? Why? And when?
- **Stage 1: Independent data extraction** [10 min]
  - How did you conduct data extraction activity?
  - List pros and cons of extracting data using color-coding annotation require you to associate evidence with defined themes or codes
  - Which limitations have you found when highlighting?
  - To what extent metadata extraction helps you in being more efficient in your task?
  - To what extent transparent storage in spreadsheets helps you in being more efficient in your labor?
  - Do you look up your extraction spreadsheet during independent data extraction? When? What for?
  - Did you navigate through the previously highlighted fragments before taking a classification decision?
  - How do you decide the data extraction activity is completed?
- **Stage 2: Check extracted data** [10 min]
  - How did you conduct your data extraction checking?
  - Which limitations have you found during checking?
  - Do you think that cell-color semaphore helped you to be more consistent during data extraction?

  - To what extent did the notes over cells help you to observe the decisions taken?
  - Did you use hyperlinks to coding context? If so, are they useful to trace taken decisions?

- **Stage 3: Analysis and Synthesis** [5 min]
  - How did you conduct the analysis and synthesis step? To what extent did you find useful resultant data for this stage?
  - How have you used links to the coding context in the step of analysis?
  - How did you process the resultant data (the spreadsheet)?
  - What kind of visualization have you created from the resultant spreadsheet?
  - Did you use any of the quality metrics provided in the "Audit" sheet?
  - What tools/technologies have you used for data analysis and synthesis apart from those used by *Highlight&Go* (e.g.: R)? How easy was it to import data to these tools?

- **Explore changes and future directions** [10 min]
  - In what (unforeseen) other SLR activities can be used *Highlight&Go*?
  - To what extent do you think data extraction can be automatized using annotation?

- **Closing** [5 min]
  - List pros and cons of *Highlight&Go* comparing to Spreadsheets (Excel) and PDF readers alone
  - List pros and cons of *Highlight&Go* as a tool combined with Google Sheets comparing to those as standalone tools (e.g. nVivo,...)
  - Are you planning to publish the generated spreadsheet by *Highlight&Go* (including hyperlinks to evidence)?
  - Close up asking them for final comments

Appendix B

Mark&Go confirmatory focus group guide

This section presents the focus group interview guide[1] used to facilitate the instructor labor to moderate, posing questions and adding prompting follow-up questions to encourage the discussion between practitioners[2]:

- **Introduction** [10 min]
  - Welcome and ask for permission to record
  - Introduction to the purpose of the focus group
  - Ask participants a short introduction about their teaching experience and context, where *Mark&Go* has been used
- **General questions** [20 min]
  - Workflow: How does *Mark&Go* fit your usual marking workflow? Did you need to change anything from your usual workflow? If so, what and why?
  - Feedback quality: Do you perceive that feedback quality has increased? Do you feel that your feedback has better quality using *Mark&Go* compared to Moodle alone?
  - Limitations: Were you able to use the tool to fully conduct your marking activity using *Mark&Go*? Do you think that there are some assignment types or contexts that *Mark&Go* works better for?
  - Satisfaction and intention to use: What is your overall satisfaction with the tool? Are you planning to use it the next time you have to assess students' assignments?

[1] The full version can be found in: https://rebrand.ly/markAndGoCompleteFocusGroupGuide

[2] Additionally we have shared a short version of the guide with practitioners before the meeting: https://rebrand.ly/markAndGoSharedFocusGroupGuide

- **Explore changes and future directions** [15min]
  - Do you think that *Mark&Go* can be useful in any other contexts?
  - What other platforms would be interesting to support in the tool?
- **Mechanism utility**: we questioned participants to rank *Mark&Go*'s mechanism of utility for quality feedback [10min]
  - Color-coding highlighter based on the evaluation rubric
  - Look-back commenting based on student's previous assignments
  - Grading facilities: automatic translation of marks to Moodle
  - Time estimations
  - Resumption facility
- **Correction stage** [20 min]
  - Do time estimations help you to be more timely, plan yourself better, to better predict how much labor is pending, and reduce your procrastination?
  - How common is it for you to assess the assignments in more than one sitting? Does the resumption facility help you to resume your assessment activity in those cases?
  - Highlighting:
    - List pros and cons of assessing using color-coding annotation require you to associate evidence with rubrics
    - To what extent have you modified the evaluation rubric during a correction?
    - Which limitations have they found when highlighting?
    - How can this mechanism be improved?
  - Commenting:
    - List pros and cons of look-back commenting to provide personalized feedback
    - Do comments reuse make you more timely when providing feedback comments?
    - Which limitations have they found when commenting?
    - How can this mechanism be improved?
  - Marking:
    - How have you used the sidebar for marking? Did you highlight and mark at the same time?
    - In what scenarios or type of assignment did you use navigation facilities to decide a mark?
    - List pros and cons of look-back commenting to provide personalized feedback

    - Which limitations have they found when marking?
    - How can this mechanism be improved?

- **Reporting stage** [10 min]
  - Does the automatic feedback translation to Moodle increase timely feedback?
  - Is the generated textual report complete?
  - Have you enabled the automatic submission of annotated files? Do you find it useful for your students?
  - Which limitations have you found when marking?
  - How can this mechanism be improved?

- **Closing** [5 min]
  - Feedback access: Did you feel that students have increased access to their feedback when correcting using *Mark&Go*? Are you aware of students that have installed *Mark&Go* to review the feedback in the context?
  - Close up asking them for final comments

Appendix C

Annotation projects' variants configuration

The following section shows the features selected to derive the web annotation products described as the main examples in this Thesis: *Highlight&Go*, *Mark&Go* and *Review&Go*. This configurations are done using *pure::variants* and derived as presented in Section 7.3.

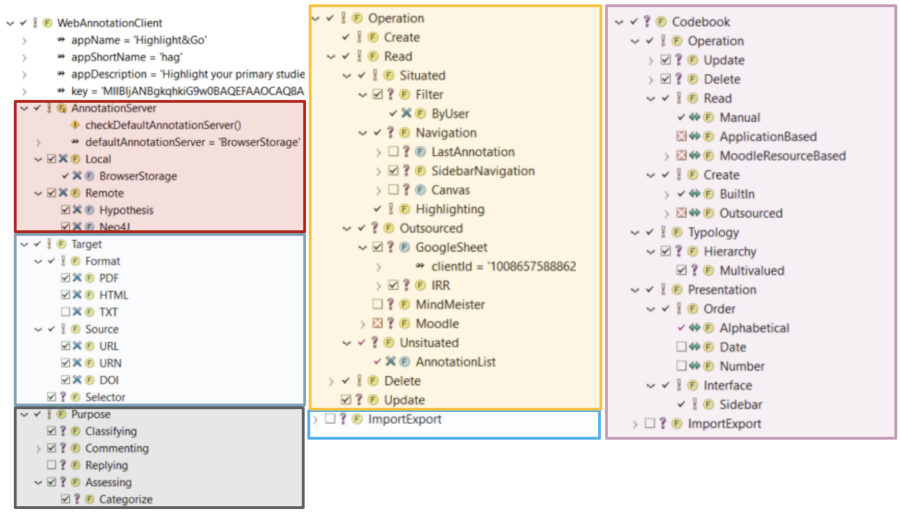


Figure C.1: Selected features to derive *Highlight&Go* product in WACline.

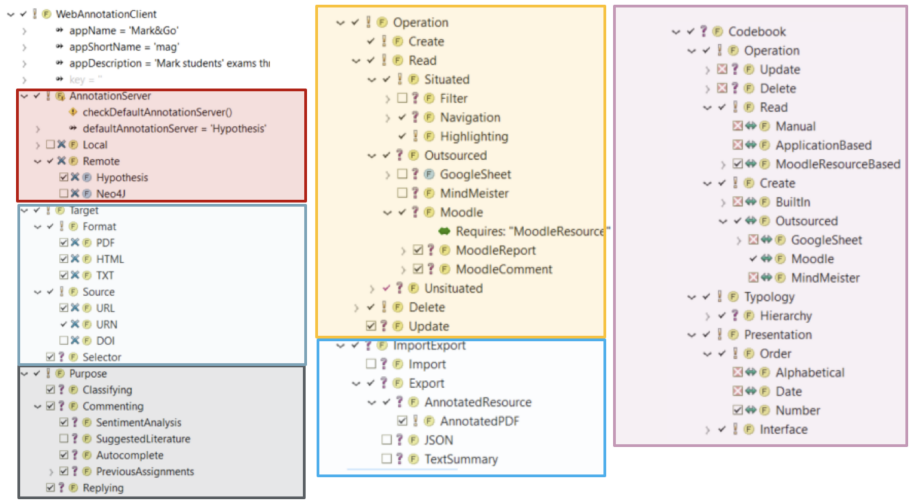


Figure C.2: Selected features to derive *Mark&Go* product in WACline.

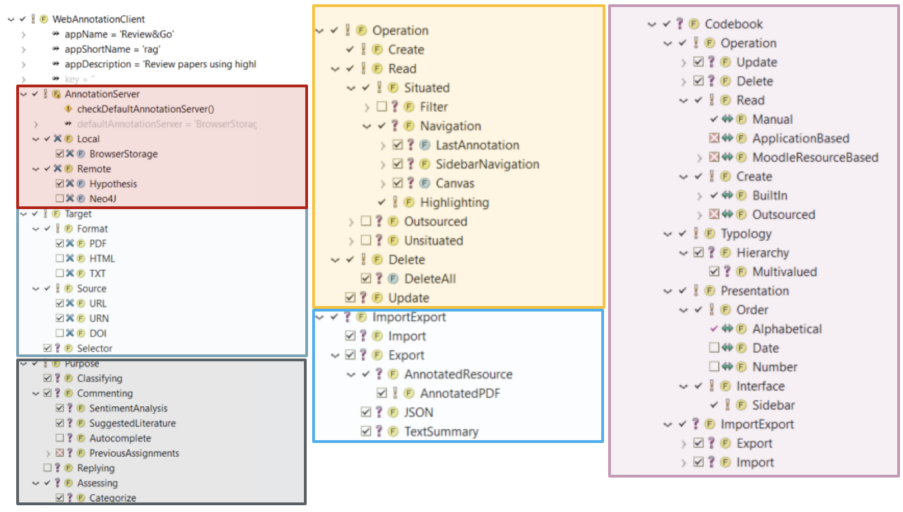


Figure C.3: Selected features to derive *Review&Go* product in WACline.

Appendix D

Review draft automatically generated out of reviewer annotations

```
<Summarize the work>

STRENGTHS:
- the proposed solution is clear and convincing.
 * (Page 6): "It is available for download at the Chrome Web Store".
The availability of the artifact is a plus.
- the artifact has been compared with extant solutions.
 * (Page 12): "Is Review&Go perceived to be better than
  conducting the review through Acrobat Reader". The
  comparison with Acrobat Reader is pertinent.

MINOR WEAKNESSES:
There is a minor point that should be clarified. The paper
seems to overlook the 'why' and focus too much on the 'what'.
 * (Page 1): "Different causes can be blamed for this situation:
 (1) lack of transparency in the process [18,5], (2) lack of
 agreement about what constitutes good reviewing [18,16,24,8],
 (3) lack of skills and re-viewing experience [11,8], or (4)
 lack of time". The problem should be analyzed in more detail.
 I would encourage the authors to look at the following papers: [1]

TYPOS:
- (Page 1): "raison d'etre"

REFERENCES:
```

[1] Richard Baskerville, Abayomi Baiyere, Shirley Gregor, Alan R. Hevner, Matti Rossi: Design Science Research Contributions - Finding a Balance between Artifact and Theory. (2018)

<Comments to editors>

Bibliography

[ABA+19] Apostolos Ampatzoglou, Stamatia Bibi, Paris Avgeriou, Marijn Verbeek, and Alexander Chatzigeorgiou. Identifying, categorizing and mitigating threats to validity in software engineering secondary studies. *Information and Software Technology*, 106:201–230, 2019.

[ABKS13] Sven Apel, Don Batory, Christian Kästner, and Gunter Saake. *Feature-oriented software product lines*. Springer, 2013.

[Aca20] Enago Academy. Experts' take on peer review evaluation (pre). `https://www.enago.com/academy/experts-take-on-peer-review-evaluation/`, Sep 2020. Accessed 27 Jan 2022.

[Ada14] Mireilla Bikanga Ada. Using myfeedback application to statistically investigate students' feedback access. In *Proceedings ELMAR-2014*, pages 1–4. IEEE, 2014.

[ADNF07] Maristella Agosti, Giorgio Maria Di Nunzio, and Nicola Ferro. The importance of scientific data curation for evaluation campaigns. In *International DELOS Conference*, pages 157–166. Springer, 2007.

[Ado22] Adobe. Importing and exporting comments. `https://helpx.adobe.com/acrobat/using/importing-exporting-comments.html`, 2022. Accessed 27 Jan 2022.

[AG10] David S Ackerman and Barbara L Gross. Instructor feedback: How much do students really want? *Journal of Marketing Education*, 32(2):172–181, 2010.

[AIMD21] Maider Azanza, Arantza Irastorza, Raul Medeiros, and Oscar Díaz. Onboarding in software product lines: Concept maps as welcome guides. In *43rd IEEE/ACM International Conference on Software Engineering: Software Engineering Education and Training, ICSE (SEET) 2021, Madrid, Spain, May 25-28, 2021*, pages 122–133. IEEE, 2021.

[AKKS18] Sophie Abel, Kirsty Kitto, Simon Knight, and Simon Buckingham Shum. Designing personalised, automated feedback to develop students' research writing skills. Technical report, University of Technology Sydney, 2018.

[Ako18] Haldun Akoglu. User's guide to correlation coefficients. *Turkish Journal of Emergency Medicine*, 18(3):91–93, sep 2018.

[AMD+19] Lenita Martins Ambrósio, Phillipe Marques, José Maria N. David, Regina Braga, Mário Antonio Ribeiro Dantas, Victor Ströele, and Fernanda Campos. 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 (CSCWD)*, pages 39–44, 2019.

[AMD21] Maider Azanza, Leticia Montalvillo, and Oscar Díaz. 20 years of industrial experience at splc: a systematic mapping study. In *Proceedings of the 25th ACM International Systems and Software Product Line Conference-Volume A*, pages 172–183, 2021.

[AN07] Morgan Ames and Mor Naaman. Why we tag: Motivations for annotation in mobile and online media. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, page 971–980, New York, NY, USA, 2007. Association for Computing Machinery.

[And20] Jacqulyn Machardy Anderson. Addressing Novice Coding Patterns: Evaluating and Improving a Tool for Code Analysis and Feedback. Technical report, University of Utah, 2020.

[ANS03] ANSI. ISO/IEC 9899:2018 - Programming languages — C, 2003.

[ASS18] Hosam Al-Samarraie and Noria Saeed. A systematic review of cloud computing tools for collaborative learning: Opportunities and challenges to the blended-learning environment. *Computers and Education*, 124:77–91, sep 2018.

[ATFM01] Pierre America, Steffen Thiel, Stefan Ferber, and Martin Mergel. Introduction to domain analysis. *ESAPS Project*, 2001.

[AZCHH17] Ahmed Al-Zubidy, Jeffrey C. Carver, David P. Hale, and Edgar E. Hassler. Vision for SLR tooling infrastructure: Prioritizing value-added requirements. *Information and Software Technology*, 91:72–81, nov 2017.

[BBC+13] Shannon Bradshaw, Dan Brickley, Leyla Jael García Castro, Timothy Clark, Timothy Cole, Phil Desenne, Anna Gerber, Antoine Isaac, Jacob Jett, Thomas Habing, Bernhard Haslhofer, Sebastian Hellmann, Jane Hunter, Randall Leeds, Andrew Magliozzi, Bob Morris, Paul Morris, Jacco van Ossenbruggen, Stian Soiland-Reyes, James Smith, and Dan Whaley. Open Annotation Data Model: Community Draft, 2013.

[BBW13] Steven A Burr, Elizabeth Brodier, and Simon Wilkinson. Delivery and use of individualised feedback in large class medical teaching. *BMC medical education*, 13(1):1–7, 2013.

[BCBK17] Souvik Barat, Tony Clark, Balbir Barn, and Vinay Kulkarni. A model-based approach to systematic review of research literature. In *Proceedings of the 10th Innovations in Software Engineering Conference on - ISEC '17*, pages 15–25, New York, New York, USA, 2017. ACM Press.

[BCR94] Victor Basili, Gianluigi Caldiera, and H Dieter Rombach. Goal Question Metric (GQM) Approach. In *Encyclopedia of Software Engineering*. McGraw-Hill, 1994.

[BE11] George Brown and Sarah Edmunds. *Doing pedagogical research in engineering*. Engineering Centre for Excellence in Teaching and Learning, 2011.

[Beu19] Danilo Beuche. Industrial variant management with pure::variants. In *ACM International Conference Proceeding Series*, volume B, pages 1–3, New York, New York, USA, sep 2019. Association for Computing Machinery.

[Bik14] Mireilla Bikanga Ada. Using MyFeedBack application to statistically investigate students' feedback access. In *Proceedings ELMAR-2014*, 2014.

[BJBCL18] José L. Barros-Justo, Fabiane B.V. Benitti, and Ania L. Cravero-Leal. Software patterns and requirements engineering activities in real-world settings: A systematic mapping study. *Computer Standards and Interfaces*, 58:23–42, may 2018.

[BKB+07] Pearl Brereton, Barbara A Kitchenham, David Budgen, Mark Turner, and Mohamed Khalil. Lessons from applying the systematic literature review process within the software engineering domain. *Journal of systems and software*, 80(4):571–583, 2007.

[BKW16] Fabian Beck, Sebastian Koch, and Daniel Weiskopf. Visual Analysis and Dissemination of Scientific Literature Collections with SurVis. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):180–189, 2016.

[BM13] David Boud and Elizabeth Molloy. Rethinking models of feedback for learning: The challenge of design. *Assessment and Evaluation in Higher Education*, 38(6):698–712, 2013.

[BMB17] Andre Breitenfeld and Claudia Müller-Birn. A State-of-the-Art Review of Semantic Annotation Tools. Technical report, Freie Universität, 2017.

[BMSD18] Claudio Bustos Navarrete, María Gabriela Morales Malverde, Pedro Salcedo Lagos, and Alejandro Díaz Mujica. Buhos: A web-based systematic literature review management software. *SoftwareX*, 7:360–372, jan 2018.

[BP00] Allan K. Blunt and Timothy A. Pychyl. Task aversiveness and procrastination: A multi-dimensional approach to task aversiveness across stages of personal projects. *Personality and Individual Differences*, 28(1):153–167, jan 2000.

[Bro17] Susan M Brookhart. *How to give effective feedback to your students*. ASCD, 2017.

[BZ09] Muhammad Ali Babar and He Zhang. Systematic literature reviews in software engineering: Preliminary results from interviews with researchers. In *2009 3rd International Symposium on Empirical Software Engineering and Measurement*, pages 346–355. IEEE, 2009.

[CA16] Martha Ann Carey and Jo-Ellen Asbury. *Focus group research*, volume 9. Routledge, 2016.

[Cam05] Alistair Campbell. Application of ICT and rubrics to the assessment process where professional judgement is involved: The features of an e-marking tool. *Assessment and Evaluation in Higher Education*, 30(5):529–537, 2005.

[Car06] David Carless. Differing perceptions in the feedback process. *Studies in Higher Education*, 31(2):219–233, 2006.

[Cat] Catma team - Universität Hamburg. CATMA. `https://catma.de/`. Accessed 27 Jan 2022.

[Cau20] Mike Caulfield. Digipo: The digital polarization initiative, 2020.

[CBC+21] Anderson Pinheiro Cavalcanti, Arthur Barbosa, Ruan Carvalho, Fred Freitas, Yi-Shan Tsai, Dragan Gašević, and Rafael Ferreira Mello. Automatic feedback in online learning environments: A systematic literature review. *Computers and Education: Artificial Intelligence*, 2:100027, 2021.

[CCL06] Ajay Chakravarthy, Fabio Ciravegna, and Vitakeska Lanfranchi. Cross-media document annotation and enrichment. 2006.

[CCW16] José María Cavanillas, Edward Curry, and Wolfgang Wahlster. *New horizons for a data-driven economy: a roadmap for usage and exploitation of big data in Europe*. Springer Nature, 2016.

[CD11] Daniela S. Cruzes and Tore Dybä. Research synthesis in software engineering: A tertiary study. In *Information and Software Technology*, volume 53, pages 440–455. Elsevier, may 2011.

[CDFF+16] K B Cohen, D Demner-Fushman, K Fort, C Grouin, L. E. Hunter, U Leser, A N Evéolev´ Nevéol, M Neves, and P Zweigenbaum. Towards the last annotation tool. Technical report, linkedannotation, 2016.

[CGV+17] Sarven Capadisli, Amy Guy, Ruben Verborgh, Christoph Lange, Sören Auer, and Tim Berners-Lee. Decentralised authoring, annotations and notifications for a read-write web with dokieli. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 10360 LNCS, pages 469–481, 2017.

[CHHK13] Jeffrey C Carver, Edgar Hassler, Elis Hernandes, and Nicholas A Kraft. Identifying Barriers to the Systematic Literature Review Process. In *2013 ACM / IEEE International Symposium on Empirical Software Engineering and Measurement*, pages 203–212, 2013.

[CKK21] Khalil Chehab, Anis Kalboussi, and Ahmed Hadj Kacem. Study of healthcare annotation systems. *International Journal of E-Health and Medical Communications (IJEHMC)*, 12(3):74–89, 2021.

[Cla10] Maxine Clarke. Reducing the peer-reviewer's burden. `http://blogs.nature.com/peer-to-peer/2010/05/reducing_the_peerreviewers_bur_1.html`, 2010. Accessed 27 Mar 2022.

[Cle01] Paul Clements. Control Channel Toolkit : A Software Product Line Case Study. Technical Report September, Defense Technical Information Center, 2001.

[CM05] Scott Carter and Jennifer Mankoff. When participants do the capturing: the role of media in diary studies. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 899–908, 2005.

[CM20] Federico Caria and Brigitte Mathiak. Annotation in Digital Humanities. In *Digital Cultural Heritage*, pages 39–50. Springer, 2020.

[CMP+14] Juan Miguel Cejuela, Peter McQuilton, Laura Ponting, Steven J Marygold, Raymund Stefancsik, Gillian H Millburn, and Burkhard Rost. Tagtog: Interactive and text-mining-assisted annotation of gene mentions in PLOS full-text articles. *Database*, 2014:1–8, 2014.

[CN02] Paul Clements and Linda Northrop. *Software product lines*. Addison-Wesley Boston, 2002.

[COG+11] Paolo Ciccarese, Marco Ocana, Leyla Jael Garcia Castro, Sudeshna Das, and Tim Clark. An open annotation ontology for science on web 3.0. *Journal of Biomedical Semantics*, 2(2):1–24, may 2011.

[CS90] Juliet M Corbin and Anselm Strauss. Grounded theory research: Procedures, canons, and evaluative criteria. *Qualitative sociology*, 13(1):3–21, 1990.

[CSRC13] Paolo Ciccarese, Stian Soiland-Reyes, and Tim Clark. Web Annotation as a First Class Object. *IEEE Internet Comput.*, pages 71–75, 2013.

[DA15] Oscar Díaz and Cristóbal Arellano. The augmented web: Rationales, opportunities, and challenges on browser-side transcoding. *ACM Trans. Web*, 9(2):8:1–8:30, 2015.

[dABMN+07] Jorge Calmon de Almeida Biolchini, Paula Gomes Mian, Ana Candida Cruz Natali, Tayana Uchôa Conte, and Guilherme Horta Travassos. Scientific research ontology to support systematic review in software engineering. *Advanced Engineering Informatics*, 21(2):133–151, 2007.

[Dag00] James C. Dager. Cummins's Experience in Developing a Software Product Line Architecture for Real-time Embedded Diesel Engine Controls. In *Software Product Lines*, pages 23–45. Springer US, 2000.

[DdOH20] Aitor Díaz de Otazu Hernando. Desarrollo de una aplicación para el análisis de sentencias judiciales utilizando la línea de productos software wacline. `http://hdl.handle.net/10810/48781`, 2020. Accessed 27 Jan 2022.

[DG17] Lia M. Daniels and Mark J. Gierl. The impact of immediate test score reporting on university students' achievement emotions in the context of computer-based multiple-choice exams. *Learning and Instruction*, 52:27–35, dec 2017.

[DHM+18] Phillip Dawson, Michael Henderson, Paige Mahoney, Michael Phillips, Tracii Ryan, David Boud, and Elizabeth Molloy. What makes for effective feedback: staff and student perspectives. *Assessment and Evaluation in Higher Education*, 2938:1–12, 2018.

[DJtBAvD21] Kim Dirkx, Desirée Joosten-ten Brinke, Jorik Arts, and Migchiel van Diggelen. In-text and rubric-referenced feedback: Differences in focus, level, and function. *Active Learning in Higher Education*, 22(3):189–201, 2021.

[DKUT09] John Daniel, Asha Kanwar, and Stamenka Uvalić-Trumbić. Breaking higher education's iron triangle: Access, cost, and quality. *Change: The Magazine of Higher Learning*, 41(2):30–35, 2009.

[DMA19] Oscar Díaz, Haritz Medina, and Felipe I. Anfurrutia. Coding-Data Portability in Systematic Literature Reviews. In *Proceedings of the Evaluation and Assessment on Software Engineering*, pages 178–187, 2019.

[EK18] Colin Elman and Diana Kapiszewski. The qualitative data repository's annotation for transparent inquiry (ati) initiative. *PS: Political Science & Politics*, 51(1):3–6, 2018.

[EpA17] EJournalPress, Hypothes.is project, and American Geophysical Union. AGU Launches Hypothesis to Facilitate Peer Review. `https://web.hypothes.is/blog/agu-launches-hypothesis-to-facilitate-peer-review/`, 2017. Accessed 27 Jan 2022.

[EPPC21] Jorge Echeverría, Francisca Pérez, José Ignacio Panach, and Carlos Cetina. An empirical study of performance using clone & own and software product lines in an industrial context. *Information and Software Technology*, 130:106444, 2021.

[Eve18] Jeanine C Evers. Current Issues in Qualitative Data Analysis Software (QDAS): A User and Developer Perspective. *The Qualitative Report*, 23(13):61–73, 2018.

[FC20] Katia R. Felizardo and Jeffrey C. Carver. *Automating Systematic Literature Review*, pages 327–355. Springer International Publishing, Cham, 2020.

[FLT03] Helen Finch, Jane Lewis, and Caroline Turley. Focus groups. *Qualitative research practice: A guide for social science students and researchers*, 2:211–242, 2003.

[FMS+17] Wolfram Fenske, Jens Meinicke, Sandro Schulze, Steffen Schulze, and Gunter Saake. Variant-preserving refactorings for migrating cloned products to a product line. In *2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pages 316–326. IEEE, 2017.

[For08] Martin Forsey. Ethnographic interviewing: From conversation to published text. In *How to do educational ethnography*, pages 57–75. The Tufnell Press, 2008.

[For20] CC2020 Task Force. *Computing Curricula 2020: Paradigms for Global Computing Education*. Association for Computing Machinery, New York, NY, USA, 2020.

[FP10] Jeremy Fox and Owen L Petchey. Pubcreds: fixing the peer review process by "privatizing" the reviewer commons. *The Bulletin of the Ecological Society of America*, 91(3):325–333, 2010.

[Gar20] Xabier Garmendia. Feature-based software development: a case for web annotation-based tools. Master's thesis, Facultad de Informática, Universidad de Murcia, 2020.

[GCGdJGA+19] Joaquín Gayoso-Cabada, María Goicoechea-de Jorge, Mercedes Gómez-Albarrán, Amelia Sanz-Cabrerizo, Antonio Sarasa-Cabezuelo, and José-Luis Sierra. Ontology-Enhanced Educational Annotation Activities. *Sustainability*, 11(16):4455, aug 2019.

[GCSCS13] Joaquín Gayoso-Cabada, Amelia Sanz-Cabrerizo, and José Luis Sierra. @note: An electronic tool for academic readings. *ACM International Conference Proceeding Series*, pages 0–3, 2013.

[GCSCS18] Joaquín Gayoso-Cabada, Antonio Sarasa-Cabezuelo, and José-Luis Sierra. Document Annotation Tools. In *Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality - TEEM'18*, volume 7, pages 889–895, New York, New York, USA, 2018. ACM Press.

[GF17] Vahid Garousi and Michael Felderer. Experience-based guidelines for effective and efficient data extraction in systematic reviews in software engineering. In *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering*, pages 170–179. ACM, 2017.

[GH13a] Barry Gibson and Jan Hartman. Introduction of Rediscovering Grounded Theory. In *Rediscovering Grounded Theory*, pages 1–24. Sage, 2013.

[GH13b] T. Grandon Gill and Alan R Hevner. A fitness-utility model for design science research. *ACM Transactions on Management Information Systems*, 4(2), 2013.

[GH13c] Shirley Gregor and Alan R Hevner. Positioning and Presenting Design Science Research for Maximum Impact. *MIS Quarterly*, 37(2):337–355, 2013.

[GHJvB17] Alex Gerdes, Bastiaan Heeren, Johan Jeuring, and L Thomas van Binsbergen. Ask-elle: an adaptable programming tutor for haskell giving automated feedback. *International Journal of Artificial Intelligence in Education*, 27(1):65–100, 2017.

[Gib05] Graham Gibbs. *Improving the quality of student learning*. University of South Wales (United Kingdom), 2005.

[GKS20] Shirley Gregor, L Chandra Kruse, and Stefan Seidel. The anatomy of a design principle. *Journal of the Association for Information Systems*, 2020.

[GMC+15] James Galipeau, David Moher, Craig Campbell, Paul Hendry, D William Cameron, Anita Palepu, and Paul C Hébert. A systematic review highlights a knowledge gap regarding the effectiveness of health-related training programs in journalology. *Journal of Clinical Epidemiology*, 68(3):257–265, 2015.

[Gra] Richard P. Grant. On peer review. `http://occamstypewriter.org/rpg/2010/04/15/on_peer_review/`. Accessed 27 Jan 2022.

[GRAJ14] Ernesto Galbán-Rodríguez and Ricardo Arencibia-Jorge. Editorials and cascading peer review. *European Science Editing*, 40(2):34–35, 2014.

[GSA18] Hajar Ghadirian, Keyvan Salehi, and Ahmad Fauzi Mohd Ayub. Social annotation tools in higher education: a preliminary systematic review. *International Journal of Learning Technology*, 13(2):130–162, 2018.

[GST+02] Jim Gray, Alexander S Szalay, Ani R Thakar, Christopher Stoughton, et al. Online scientific data curation, publication, and archiving. In *Virtual observatories*, volume 4846, pages 103–107. International Society for Optics and Photonics, 2002.

[Ham12] Irene Hames. Peer review in a rapidly evolving publishing landscape. In *Academic and professional publishing*, pages 15–52. Elsevier, 2012.

[Ham14] Hamish Cunningham et al. Developing Language Processing Components with GATE Version 8 (a User Guide). http://gate.ac.uk/userguide, 2014. Accessed 27 Jan 2022.

[HBKV21] Leigh-Anne Hepburn, Madeleine Borthwick, Jane Kerr, and Andrey Vasnev. A strategic framework for delivering ongoing feedback at scale. *Assessment & Evaluation in Higher Education*, 0(0):1–13, 2021.

[HCHAZ16] Edgar Hassler, Jeffrey C Carver, David Hale, and Ahmed Al-Zubidy. Identification of SLR tool needs - Results of a community workshop. *Information and Software Technology*, 70:122–129, 2016.

[Her12] Rosario Hernández. Does continuous assessment in higher education support student learning? *Higher Education*, 64(4):489–502, oct 2012.

[Hev07] Alan R Hevner. A three cycle view of design science research. *Scandinavian journal of information systems*, 19(2):4, 2007.

[HHS01] Richard Higgins, Peter Hartley, and Alan Skelton. Getting the message across: the problem of communicating assessment feedback. *Teaching in higher education*, 6(2):269–274, 2001.

[HKM06] William A. Hetrick, Charles W. Krueger, and Joseph G. Moore. Incremental return on incremental investment: Engenio's transition to software product line practice. In *Proceedings of the Conference on Object-Oriented Programming Systems, Languages, and Applications, OOPSLA*, volume 2006, pages 798–804, New York, New York, USA, 2006. ACM Press.

[HP13] Marina Harvey PhD. Setting the standards for sessional staff: quality learning and teaching. *Journal of University Teaching & Learning Practice*, 10(3):4, 2013.

[HPCWA18] Alan Hevner, Nicolas Prat, Isabelle Comyn-Wattiau, and Jacky Akoka. A pragmatic approach for identifying and managing design science research goals and evaluation criteria. In *AIS SIGPrag Pre-ICIS workshop on" Practice-based Design and Innovation of Digital Artifacts"*, 2018.

[HT07] John Hattie and Helen Timperley. The Power of Feedback. *Review of Educational Research*, 77(1):81–112, mar 2007.

[Hux07] Mark Huxham. Fast and effective feedback: are model answers the answer? *Assessment & Evaluation in Higher Education*, 32(6):601–611, dec 2007.

[HWH20] Kathryn Haughney, Shawnee Wakeman, and Laura Hart. Quality of feedback in higher education: A review of literature. *Education Sciences*, 10(3), 2020.

[Hyp19] Hypothes.is. Hypothesis Releases Gradebook Integration for Blackboard, Moodle, D2L and More - Hypothesis. `https://rebrand.ly/hypothesisLTIBlog`, 2019. Accessed 27 Jan 2022.

[IJG08] John Immerwahr, Jean Johnson, and Paul Gasbarra. The iron triangle: College presidents talk about costs, access, and quality. Technical report, Virginia Tech, 2008.

[IMS10] IMS Global Learning Consortium. IMS Basic Learning Tools Interoperability, 2010.

[IV09] Juhani Iivari and John R Venable. Action research and design science research-seemingly similar but decisively dissimilar. In *ECIS 2009 PROCEEDINGS*, 2009.

[Iva17] Ada Ivanova. 6 of the Best Google Chrome Extensions to Annotate Text on the Web. `https://www.maketecheasier.com/google-chrome-extensions-annotate-text-on-the-web/`, 2017. Accessed 27 Jan 2022.

[JP14] Paul Johannesson and Erik Perjons. *An introduction to design science*, volume 9783319106. Springer, 2014.

[K+99] Daniel Kahneman et al. Objective happiness. *Well-being: The foundations of hedonic psychology*, 3(25):1–23, 1999.

[KBB15] B. Kitchenham, D. Budgen, and O. P. Brereton. *Evidence-Based Software Engineering and Systematic Reviews*. CRC press, 2015.

[KC07] Barbara Kitchenham and Stuart Charters. Guidelines for performing Systematic Literature reviews in Software Engineering Version 2.3. *Engineering*, 45(4ve):1051, 2007.

[KD13] George Kamberelis and Greg Dimitriadis. *Focus groups: From structured interviews to collective conversations*. Routledge, 2013.

[Kit96] Barbara Kitchenham. DESMET: A method for evaluating Software Engineering methods and tools. *Computing & Control Engineering Journal*, 8(3):120–126., 1996.

[KK01] José Kahan and Marja-Ritta Koivunen. Annotea: an open rdf infrastructure for shared web annotations. In *Proceedings of the 10th international conference on World Wide Web*, pages 623–632, 2001.

[KKPW21] Sebastian Karcher, Dessislava Dessi Kirilova, Christiane Pagé, and Nic Weber. How data curation enables epistemically responsible reuse of qualitative data. *The Qualitative Report*, 26(6):1996–2010, 2021.

[KKR08] Robert M. Klassen, Lindsey L. Krawchuk, and Sukaina Rajani. Academic procrastination of undergraduates: Low self-efficacy to self-regulate predicts higher levels of procrastination. *Contemporary Educational Psychology*, 33(4):915–931, oct 2008.

[KMK16] Anis Kalboussi, Omar Mazhoud, and Ahmed Hadj Kacem. Functionalities provided by annotation systems for learners in educational context: An overview. *International Journal of Emerging Technologies in Learning*, 11(2):4–11, 2016.

[KPA11] Ricardo Kawase, George Papadakis, and Fabian Abel. Generating resource profiles by exploiting the context of social annotations. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 7031 LNCS, pages 289–304, 2011.

[KPSY07] Minseong Kim, Sooyong Park, Vijayan Sugumaran, and Hwasil Yang. Managing requirements conflicts in software product lines: A goal and scenario based approach. *Data and Knowledge Engineering*, 61(3):417–432, jun 2007.

[Kru01] Charles W. Krueger. Easing the transition to software mass customization. In *International Workshop on Software Product-Family Engineering*, pages 282–293. Springer, 2001.

[KTV18] Akrivi Krouska, Christos Troussas, and Maria Virvou. Social Annotation Tools in Digital Learning: A Literature Review, jul 2018.

[Kun13] Sumon Kunwongse. Peer feedback, benefits and drawbacks. *Thammasat Review*, 16(3):277–288, 2013.

[LAFM16] Nicola Lettieri, Antonio Altamura, Armando Faggiano, and Delfina Malandrino. A computational approach for the experimental study of eu case law: analysis and implementation. *Social Network Analysis and Mining*, 6(1):1–17, 2016.

[Lar11] Robert S Laramee. How to Read a Visualization Research Paper. *IEEE Computer Graphics and Applications*, pages 78–82, 2011.

[LC18] Siobhan Lynam and Moira Cachia. Students' perceptions of the role of assessments at higher education. *Assessment & Evaluation in Higher Education*, 43(2):223–234, 2018.

[LFF19] Sara Laybourn, Anne C. Frenzel, and Thomas Fenzl. Teacher Procrastination, Emotions, and Stress: A Qualitative Study. *Frontiers in Psychology*, 10(October), 2019.

[LFH17] Jonathan Lazar, Jinjuan Heidi Feng, and Harry Hochheiser. *Research methods in human-computer interaction*. Morgan Kaufmann, 2017.

[LKL02] Kwanwoo Lee, Kyo C. Kang, and Jaejoon Lee. 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)*, volume 2319, pages 62–77, 2002.

[LKL+19] Christina Lohr, Johannes Kiesel, Stephanie Luther, Johannes Hellrich, Tobias Kolditz, Benno Stein, and Udo Hahn. Continuous Quality Control and Advanced Text Segment Annotation with WAT-SL 2.0. In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 215–219. Association for Computational Linguistics (ACL), sep 2019.

[LRF11] Travis I Lovejoy, Tracey A Revenson, and Christopher R France. Reviewing manuscripts for peer-review journals: a primer for novice and seasoned reviewers. *Annals of Behavioral Medicine*, 42(1):1–13, 2011.

[LW08] Alf Lizzio and Keithia Wilson. Feedback on assessment: Students' perceptions of quality and effectiveness. *Assessment and Evaluation in Higher Education*, 33(3):263–275, 2008.

[Mas86] Imai Masaaki. *Kaizen: The key to Japan's competitive success*. McGraw-Hill Education, 1986.

[MBKB+15] Claudia Müller-Birn, Tina Klüwer, André Breitenfeld, Alexa Schlegel, and Lukas Benedix. neonion - combining human and machine intelligence. In *Proceedings of the 18th ACM Conference Companion on Computer Supported Cooperative Work & Social Computing - CSCW'15 Companion*, volume 2015-Janua, pages 223–226, New York, New York, USA, feb 2015. ACM Press.

[McE04] Elaine K McEwan. *Seven strategies of highly effective readers: using cognitive research to boost K-8 achievement*. Corwin press, 2004.

[MD16] Leticia Montalvillo and Oscar Díaz. Requirement-driven Evolution in Software Product Lines: A Systematic Mapping Study. *The Journal of Systems and Software*, pages 1–74, 2016.

[MD22] Raul Medeiros and Oscar Díaz. Assisting mentors in selecting newcomers' next task in software product lines: A recommender system approach. In *34th International Conference on Advanced Information Systems Engineering*, 2022.

[MDA18] Haritz Medina, Oscar Dıaz, and Felipe I Anfurrutia. Highlight&go: una extensión para automatizar la extracción de datos en revisiones sistemáticas de la literatura utilizando google sheets. In *JISBD2018*, 2018.

[MH19] Matthew T Mullarkey and Alan R Hevner. An elaborated action design research process model. *European Journal of Information Systems*, 28(1):6–20, 2019.

[MJHP12] Kevin Matthews, Thomas Janicki, Ling He, and Laurie Patterson. Implementation of an automated grading system with an adaptive learning component to affect student feedback and response time. *Journal of Information Systems Education*, 23(1):71–80, 2012.

[MML98] Kathleen M Macqueen and Eleanor McLellan-Lemal. Team-based codebook development : Structure , process , and agreement. *Cultural Antropology Methods*, 10(2):31–36, 1998.

[MMM98] Ali Mili, Rym Mili, and Roland T. Mittermeir. A survey of software reuse libraries. *Ann. Softw. Eng.*, 5:349–414, 1998.

[Moz22a] Mozilla Foundation. Browser Extensions - Mozilla | MDN. `https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions`, 2022. Accessed 27 Jan 2022.

[Moz22b] Mozilla Foundation. Chrome incompatibilities - Mozilla | MDN. `https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/Chrome_incompatibilities`, 2022. Accessed 27 Jan 2022.

[MRG+18] Flávio Medeiros, Márcio Ribeiro, Rohit Gheyi, Sven Apel, Christian Kästner, Bruno Ferreira, Luiz Carvalho, and Baldoino Fonseca. Discipline Matters: Refactoring of Preprocessor Directives in the #ifdef Hell. *IEEE Transactions on Software Engineering*, 44(5):453–469, may 2018.

[MS14] Felix Maringe and Nevensha Sing. Teaching large classes in an increasingly internationalising higher education environment: Pedagogical, quality and equity issues. *Higher Education*, 67(6):761–782, 2014.

[MT04] Tom Mens and Tom Tourwé. A survey of software refactoring. *IEEE Transactions on software engineering*, 30(2):126–139, 2004.

[MT17] Emma Mulliner and Matthew Tucker. Feedback on feedback practice: perceptions of students and academics. *Assessment and Evaluation in Higher Education*, 42(2):266–288, 2017.

[Nay16] Alexander Naydenov. Paperhive–a coworking hub for researchers that aims to makereading more collaborative. *Impact of Social Sciences Blog*, 2016.

[Nic10] David Nicol. From monologue to dialogue: improving written feedback processes in mass higher education. *Assessment & Evaluation in Higher Education*, 35(5):501–517, aug 2010.

[NO16] Bjoern Niehaves and Kevin Ortbach. The inner and the outer model in explanatory design theory: the case of designing electronic feedback systems. *European Journal of Information Systems*, 25(4):303–316, 2016.

[Nos17] Brian A Nosek. Center for open science: Strategic plan, Mar 2017.

[NS18] Vilmar Nepomuceno and Sergio Soares. Maintaining systematic literature reviews. In *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement - ESEM 18*, pages 1–4, New York, New York, USA, 2018. ACM Press.

[NŠ19] Mariana Neves and Jurica Ševa. An extensive review of tools for manual annotation of documents. *Briefings in Bioinformatics*, dec 2019.

[OTG+19] Annette M. O'Connor, Guy Tsafnat, Stephen B. Gilbert, Kristina A. Thayer, Ian Shemilt, James Thomas, Paul Glasziou, and Mary S. Wolfe. Still moving toward automation of the systematic review process: a summary of discussions at the third meeting of the International Collaboration for Automation of Systematic Reviews (ICASR). *Systematic Reviews*, 8(1):57, dec 2019.

[Par16] K Paradis, J. & Fendt. Annotation Studio - Digital Annotation as an Educational Approach in the Humanities and Arts, 2016.

[Pay22] Ganesh Payyanur. On peer review, 2022. Accessed 27 Jan 2022.

[PB14] Catherine Pickering and Jason Byrne. The benefits of publishing systematic quantitative literature reviews for PhD candidates and other early-career researchers. *Higher Education Research & Development*, 33(3):534–548, 2014.

[PBvdL05] Klaus Pohl, Günter Böckle, and Frank van der Linden. *Software Product Line Engineering - Foundations, Principles, and Techniques*. Springer, 2005.

[PD19] Juanan Pereira and Óscar Díaz. Using Health Chatbots for Behavior Change: A Mapping Study. *Journal of Medical Systems*, 43(5):135, 2019.

[PFMM08] Kai Petersen, Robert Feldt, Shahid Mujtaba, and Michael Mattsson. Systematic Mapping Studies in Software Engineering. In *12th International Conference on Evaluation and Assessment in Software*, pages 1–10, 2008.

[PJD+19] Abelardo Pardo, Jelena Jovanovic, Shane Dawson, Dragan Gašević, and Negin Mirriahi. Using learning analytics to scale the provision of personalised feedback. *British Journal of Educational Technology*, 50(1):128–138, jan 2019.

[PKD+12] Bernardo Pereira Nunes, Ricardo Kawase, Stefan Dietze, Gilda Helena Bernardino De Campos, and Wolfgang Nejdl. Annotation tool for enhancing e-learning courses. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 7558 LNCS, pages 51–60, 2012.

[PLCPYC16] Jose Luis Poza-Lujan, Carlos T Calafate, Juan Luis Posadas-Yague, and Juan Carlos Cano. Assessing the Impact of Continuous Evaluation Strategies: Tradeoff between Student Performance and Instructor Effort. *IEEE Transactions on Education*, 59(1):17–23, 2016.

[PPMMD17] Abelardo Pardo, Oleksandra Poquet, Roberto Martínez-Maldonado, and Shane Dawson. Provision of data-driven student feedback in la & edm. *Handbook of learning analytics*, pages 163–174, 2017.

[PRC16] Publishing Research Consortium PRC. MS Windows NT kernel description. https://www.elsevier.com/__data/assets/pdf_file/0007/655756/PRC-peer-review-survey-report-Final-2016-05-19.pdf, 2016. Accessed: 2022-01-24.

[pro15] AnnotatorJS project. AnnotatorJS - Annotating the Web. http://annotatorjs.org/, 2015. Accessed 27 Jan 2022.

[PT06] Andrew Parker and Jonathan Tritter. Focus group method and methodology: current practice and recent debate. *International Journal of Research & Method in Education*, 29(1):23–37, 2006.

[PVK15] Kai Petersen, Sairam Vakkalanka, and Ludwik Kuzniarz. Guidelines for conducting systematic mapping studies in software engineering: An update. In *Information and Software Technology*, volume 64, pages 1–18, 2015.

[Rap70] Robert N Rapoport. Three dilemmas in action research: with special reference to the tavistock experience. *Human relations*, 23(6):499–513, 1970.

[RBB+15] Heidi L Rehm, Jonathan S Berg, Lisa D Brooks, Carlos D Bustamante, James P Evans, Melissa J Landrum, David H Ledbetter, Donna R Maglott, Christa Lese Martin, Robert L Nussbaum, et al. Clingen—the clinical genome resource. *New England Journal of Medicine*, 372(23):2235–2242, 2015.

[RBB+20] Paul Ralph, Sebastian Baltes, Domenico Bianculli, Yvonne Dittrich, Michael Felderer, Robert Feldt, Antonio Filieri, Carlo Alberto Furia, Daniel Graziotin, Pinjia He, Rashina Hoda, Natalia Juristo, Barbara A. Kitchenham, Romain Robbes, Daniel Méndez, Jefferson Molleri, Diomidis Spinellis, Miroslaw Staron, Klaas-Jan Stol, Damian A. Tamburri, Marco Torchiano, Christoph Treude, Burak Turhan, and Sira Vegas. ACM SIGSOFT empirical standards. *CoRR*, abs/2010.03525, 2020.

[RBF+18] Alexander Rozental, Sophie Bennett, David Forsström, David D. Ebert, Roz Shafran, Gerhard Andersson, and Per Carlbring. Targeting procrastination using psychological treatments: A systematic review and meta-analysis, aug 2018.

[Reh20] Georg Rehm. Observations on annotations. *Annotations in Scholarly Editions and Research: Functions, Differentiation, Systematization*, page 299, 2020.

[RFK19] Tracii Ryan, Sarah French, and Gregor Kennedy. Beyond the Iron Triangle: improving the quality of teaching and learning at scale. *Studies in Higher Education*, 2019.

[RGID+09] G Rodríguez Gomez, MS Ibarra Sáiz, JM Dodero, MA Gómez Ruiz, B Gallego Noche, D Cabeza Sanchez, V Quesada Serra, and Álvaro Martínez del Val. Developing the e-Learning-oriented e-Assessment. In *V International Conference on Multimedia and Information and Communication*

Technologies in Education, pages 515–519. Formatex Lisbon, 2009.

[RMSB19] Georg Rehm, Julian Moreno-Schneider, and Peter Bourgonje. Automatic and manual web annotations in an infrastructure to handle fake news and other online media phenomena. *LREC 2018 - 11th International Conference on Language Resources and Evaluation*, pages 2416–2422, 2019.

[RR05] Donna J Reid and Fraser JM Reid. Online focus groups: An in-depth comparison of computer-mediated and conventional focus group discussions. *International journal of market research*, 47(2):131–162, 2005.

[RRG+17] Jason Rhode, Stephanie Richter, Peter Gowen, Tracy Miller, and Cameron Wills. Understanding faculty use of the learning management system. *Online Learning*, 21(3):68–86, 2017.

[Sal16] Kim Salazar. Diary studies: Understanding long-term user behaviour and experiences. `https://www.nngroup.com/articles/diary-studies/`, 2016. Accessed 22 Feb 2022.

[SBID17] Rainer Simon, Elton Barker, Leif Isaksen, and Pau De Soto Cañamares. Linked Data Annotation Without the Pointy Brackets: Introducing Recogito 2. *Journal of Map and Geography Libraries*, 13(1):111–132, 2017.

[SCY17] Robert Sanderson, Paolo Ciccarese, and Benjamin Young. W3C Annotation Model - Motivation and Purpose, 2017.

[SF18] Klaas-Jan Stol and Brian Fitzgerald. The abc of software engineering research. *ACM Trans. Softw. Eng. Methodol.*, 27(3), sep 2018.

[SH94] Bruce R. Schatz and Joseph B. Hardin. Ncsa mosaic and the world wide web: Global hypermedia protocols for the internet. *Science*, 265(5174):895–901, 1994.

[SH04] Mark Staples and Derrick Hill. Experiences adopting Software Product Line Development without a Product Line Architecture. In *Proceedings - Asia-Pacific Software Engineering Conference, APSEC*, pages 176–183, 2004.

[SHP+11] Maung K Sein, Ola Henfridsson, Sandeep Purao, Matti Rossi, and Rikard Lindgren. Action design research. *MIS quarterly*, pages 37–56, 2011.

[Shu08] Valerie J Shute. Focus on formative feedback. *Review of Educational Research*, 78(1):153–189, 2008.

[SKGA17] Arjun Singh, Sergey Karayev, Kevin Gutowski, and Pieter Abbeel. Gradescope: A fast, flexible, and fair system for scalable assessment of handwritten work. In *L@S 2017 - Proceedings of the 4th (2017) ACM Conference on Learning at Scale*, pages 81–88, 2017.

[SM15] Elizaveta Sopina and Rob McNeill. Investigating the relationship between quality, format and delivery of feedback for written assignments in higher education. *Assessment & Evaluation in Higher Education*, 40(5):666–680, 2015.

[Smi10] Richard Smith. Classical peer review: an empty gun. *Breast cancer research*, 12(4):1–4, 2010.

[SMRA+16] Antonio J. Sierra, Álvaro Martín-Rodríguez, Teresa Ariza, Javier Muñoz-Calle, and Francisco J. Fernández-Jiménez. Lti for interoperating e-assessment tools with lms. In Mauro Caporuscio, Fernando De la Prieta, Tania Di Mascio, Rosella Gennari, Javier Gutiérrez Rodríguez, and Pierpaolo Vittorini, editors, *Methodologies and Intelligent Systems for Technology Enhanced Learning*, pages 173–181, Cham, 2016. Springer International Publishing.

[SN07] Mark Staples and Mahmood Niazi. Experiences using systematic review guidelines. *Journal of Systems and Software*, 80(9):1425–1437, 2007.

[Spe87] Charles Spearman. The proof and measurement of association between two things. *The American journal of psychology*, 100(3/4):441–471, 1987.

[SS15] Jagdeep Singh and Harwinder Singh. Continuous improvement philosophy – literature review and directions. *Benchmarking: An International Journal*, 22(1):75–119, feb 2015.

[SSK19] Giancarlo Sierra, Emad Shihab, and Yasutaka Kamei. A survey of self-admitted technical debt. *Journal of Systems and Software*, 152:70–82, 2019.

[Ste07] Piers Steel. The nature of procrastination. *Psychological Bulletin*, 133(1):65–94, 2007.

[TCNK16] Paolo Tell, Jacob B Cholewa, Peter Nellemann, and Marco Kuhrmann. Beyond the Spreadsheet: Reflections on Tool Support for Literature Studies. *Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering*, pages 22:1—-22:5, 2016.

[TDG+17] Jonathan P Tennant, Jonathan M Dugan, Daniel Graziotin, Damien C Jacques, François Waldner, Daniel Mietchen, Yehia Elkhatib, Lauren B Collister, Christina K Pikas, Tom Crick, et al. A multi-disciplinary perspective on emergent and future innovations in peer review. *F1000Research*, 6, 2017.

[TFPSRH21] Antonio Tenorio-Fornés, Elena Pérez Tirador, Antonio A Sánchez-Ruiz, and Samer Hassan. Decentralizing Science: Towards an Interoperable Open Peer Review Ecosystem using Blockchain. *Information Processing & Management*, 2021.

[THB10] Monica Chiarini Tremblay, Alan R. Hevner, and Donald J Berndt. Focus Groups for Artifact Refinement and Evaluation in Design Research. *Communications of the Association for Information Systems*, 26(1):27, jun 2010.

[TK82] Louis G Tornatzky and Katherine J Klein. Innovation Characteristics And Innovation Adoption-Implementation: A Meta-Analysis Of Findings. *IEEE Transactions on Engineering Management*, EM-29(1):28–45, 1982.

[Tre17] Estelle Trengove. Peer interaction as mechanism for providing timely and accessible feedback to a large undergraduate class. *The International Journal of Electrical Engineering & Education*, 54(2):119–130, 2017.

[Ude17] Udell, Jon. Federating Annotations Using Digital Object Identifiers (DOIs). `https://web.hypothes.is/blog/dois/`, 2017. Accessed 27 Jan 2022.

[VBB06] Jan Vom Brocke and Christian Buddendick. Reusable conceptual models–requirements based on the design science research paradigm. In *Proceedings of the First International Conference on Design Science Research in Information Systems and Technology (DESRIST)*, pages 576–604. Citeseer, 2006.

[vEK18] Wendelien van Eerde and Katrin B. Klingsieck. Overcoming procrastination? A meta-analysis of intervention studies, nov 2018.

[Ven06] John Venable. A framework for design science research activities. In *Emerging Trends and Challenges in Information Technology Management: Proceedings of the 2006 Information Resource Management Association Conference*, pages 184–187. Idea Group Publishing, 2006.

[Ven15] John R Venable. Five and ten years on: Have dsr standards changed? In *International Conference on Design Science Research in Information Systems*, pages 264–279. Springer, 2015.

[VN14] Richard Van Noorden. The scientists who get credit for peer review. *Nature News*, 2014.

[VWHM20] Jan Vom Brocke, Robert Winter, Alan Hevner, and Alexander Maedche. Accumulation and Evolution of Design Knowledge in Design Science Research-A Journey Through Time and Space. *Article in Journal of the Association for Information Systems*, 21:520–544, 2020.

[W3C17] W3C Web Annotation Working Group. Web Annotation. https://www.w3.org/annotation/, 2017. Accessed 27 Jan 2022.

[War11] Mark Ware. Peer review: Recent experience and future directions. *New Review of Information Networking*, 16(1):23–53, 2011.

[Wie14] Roel J. Wieringa. *Design science methodology: For information systems and software engineering*. Springer, 2014.

[Win12] Robert Winter. Construction of situational information systems management methods. *International Journal of Information System Modeling and Design (IJISMD)*, 3(4):67–85, 2012.

[WM09] Xiao-Yue Wang and Yan Mu. 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*, volume 2, pages 144–147. IEEE, 2009.

[WM15] Mark Ware and Michael Mabe. The stm report: An overview of scientific and scholarly journal publishing. Technical report, University of Nebraska Lincoln, 2015.

[WMMR06] Roel Wieringa, Neil Maiden, Nancy Mead, and Colette Rolland. Requirements engineering paper classification and evaluation criteria: A proposal and a discussion. *Requirements Engineering*, 11(1):102–107, mar 2006.

[Wor18] Molly Worthen. The misguided drive to measure 'learning outcomes'. *New York Times*, 23, 2018.

[WRAW00] Elizabeth Walsh, Maeve Rooney, Louis Appleby, and Greg Wilkinson. Open peer review: a randomised controlled trial. *The British Journal of Psychiatry*, 176(1):47–51, 2000.

[WRD+20] Moritz Wolf, Dana Ruiter, Ashwin Geet D'Sa, Liane Reiners, Jan Alexandersson, and Dietrich Klakow. HUMAN: Hierarchical universal modular annotator. *arXiv*, oct 2020.

[YBEG15] Seid Muhie Yimam, Chris Biemann, Richard Eckart de Castilho, and Iryna Gurevych. Automatic Annotation Suggestions and Custom Annotation Layers in WebAnno. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 91–96, 2015.

[YYH+12] Yueming Sun, Ye Yang, He Zhang, Wen Zhang, and Qing Wang. Towards evidence-based ontology for supporting systematic literature review. In *16th International Conference on Evaluation & Assessment in Software Engineering (EASE 2012)*, pages 171–175, 2012.

[Zim02] Barry J Zimmerman. Becoming a self-regulated learner: An overview, 2002.

[ZNY+17] Wen Zhang, Yifan Ning, Louisa Yu Zhang, Eric Chou, Richard Boyce, and Max Sibilla. AnnotationPress, 2017.

[ZSJ20] Anneke Zuiderwijk, Rhythima Shinde, and Wei Jeng. What drives and inhibits researchers to share and use open research data? a systematic literature review to analyze factors influencing open research data adoption. *PloS one*, 15(9):e0239283, 2020.

Acknowledgements

The main outcome of your Ph.D. is not your dissertation, it's you. I must admit that during this long (and short at the same time) journey, I probably learned much more than I can put into this manuscript.

First of all, I would like to thank you, the one that is reading this thesis, even if you are reading just the acknowledgments out of curiosity, but also if you are interested in what it tells or because you have to evaluate it.

I have to thank my supervisors, Prof. Dr. Oscar Díaz and Dr. Maider Azanza. It has been a great pleasure for me to have the opportunity to work with such high-level researchers. They have not only advised me during this journey, especially when I did not know where to go, but have also put me to the limit to get the best out of me as a researcher. From a personal point of view, they treated me exceptionally well.

I would like to thank the rest of the members of the Onekin group, the ones who are still part of it, and those whom I met during these years. Thank to those who have passed thousands of hours at the *labo* or even in remote, working together, taking coffee, at lunch, or in the "*speeds*", *summits*, *houstons*, or any other strange name used for a group meetings where I learned from you: Xabier Garmendia, Raul Medeiros, Jeremias Pérez, Alejo Barcina, Pablo Luján, Itziar Otaduy, Leticia Montalvillo, Iñigo Aldalur, Cristobal Arellano, Juanan Pereira, Felipe Ibañez, Iker Azpeitia, Arantza Irastorza, Jon Iturrioz and Juanmi Lopez. Thanks for all the advice, assistance, discussions, funny moments, and for sharing those moments where I have been up, but especially for suffering those moments where I have been down. I also have to thank those who are not part of the group but have been an important part of this journey at the University, Izaro for those funny lunchtimes, Unai for spending time drinking *kafetxos* and the students who worked with me during their bachelor degree, which were linked with my thesis: Eduardo Perez, Aitor Díaz de Otazu, Gorka Arce, Iñigo Bereciartua and Unai Ahedo.

I wish to thank my colleagues at Human-Centered Computing at Freie Universität, starting with my "*willkommen buddy*" Alexa Schlegel, Maximilian Mackeprang, Jesse Josua Benjamin, Michael Tebbe, Christoph Kinkeldey, Simon David Hirsbrunner, Melanie Siering, and finishing with Prof. Dr. Claudia Müller-Birn, who opened me the doors of her research group and took care of me during my stage in Berlin. It was a pity that we spent more time online than in person, but it was a pleasure to work and spend good moments with

you all.

I cannot forget all that I learned from the real experts in web annotation, my colleagues at *Hypothes.is*: Jon Udell, Jeremy Dean, Heather Staines, Nate Angell, Dan Whaley, and Robert Knight who provide me with technical help with *Hypothes.is* but also inspired me a lot at IAnnotate conferences to improve my work. Never stop annotating!

Thanks to the University of the Basque Country for financially supporting this thesis, but also the staff working at the university in *eGela* for helping me in working *Mark&Go*.

To all my friends from my hometown and those whom I meet at the university, but also my friends of *OBT* who were always asking me "when are you going to finish your Ph.D.?". Now, I can say that "it is close to over".

I would like to express my deepest thanks to my mom Espe, for supporting me during all these years and for being that strong. Also, I have to thank my grandma for taking care of me almost to the end. And thanks to Eva, Yon, and Iraia.

Last but not least, I have to express my heartfelt gratitude to Ainhoa; thank you for all those days you have distracted me from the computer, for suffering me on those days when I have been low, and also for making me a better person.

Born in Ermua, from the very beginning of my life I've been interested in technology. Probably it came from all the time spent playing video games, where I learned how to lose and try it again to finally reach a victory.

Not all of them were games, I also spent a lot of time swimming as a way to evade technology but also part of the world. Underwater there is no way to hear anything.

I started my journey as a computer scientist in a professional training school, to later moved to the university. So during my life, I combine theoretical and practical experiences. I'm passionate about learning how things work, but also how to put them into practice.

Summary

Annotation is a social behavior that helps to mediate reading-writing interaction by conveying information, adding comments, and inspiring conversation in web documents. It is used in areas from Social Sciences and Humanities, Journalism Investigation, Biological Sciences or Education, just to name a few. Annotation activities are heterogeneous, where end-users have very different requirements for creating, modifying, and reusing annotations. To facilitate reuse and interoperability, several attempts have been made during the last decades to standardize web annotations. W3C Annotation recommendations published in 2017 provide a framework for annotation representation and transport. However, there is still a gap in how annotation clients (tools and user interfaces) are developed, making developers reimplement common functionalities (i.e., highlighting, commenting, storing,...) to create their customized annotation tool.

This thesis aims to provide a reuse platform for the development of web annotation tools for review. It operationalizes this vision through a Software Product Line (WACline) that allow developers to create custom web annotation browser extensions. We reach a family of annotation clients that gives support for three reviewing practices: systematic literature review data extraction (Highlight&Go), students' assignments review in higher education (Mark&Go), and conference and journals peer-review (Review&Go). For each of the review contexts, an evaluation with real stakeholders has been conducted to validate efficiency and effectiveness improvements brought by customized annotation tools.