



HAL
open science

Interaction substrates and instruments for interaction histories

Alexandre Battut

► **To cite this version:**

Alexandre Battut. Interaction substrates and instruments for interaction histories. Human-Computer Interaction [cs.HC]. Université Paris-Saclay, 2024. English. NNT : 2024UPASG026 . tel-04626322

HAL Id: tel-04626322

<https://theses.hal.science/tel-04626322v1>

Submitted on 26 Jun 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Interaction Substrates and Instruments for Interaction Histories

Substrats et Instruments Interactifs pour les Historiques de l'Interaction

Thèse de doctorat de l'université Paris-Saclay

École doctorale n°580 :

Sciences et technologies de l'information et de la communication (STIC)

Spécialité de doctorat : Informatique

Graduate School : Informatique et sciences du numérique

Référent : Faculté des sciences d'Orsay

Thèse préparée dans l'unité de recherche **Laboratoire interdisciplinaire des sciences
du numérique** (Université Paris-Saclay, CNRS),
sous la direction de **Michel BEAUDOUIN-LAFON**, Professeur des universités

Thèse soutenue à Paris-Saclay, le 3 juin 2024, par

Alexandre BATTUT

Composition du Jury

Membres du jury avec voix délibérative

Petra ISENBERG

Directrice de recherche, Université Paris-Saclay - Inria

Présidente

Tom GROSS

Full Professor (équivalent HDR), University of Bamberg

Rapporteur & Examineur

Myriam LEWKOWICZ

Professeure des universités, Université de Technologie de Troyes

Rapporteur & Examinatrice

Clemens NYLANDSTED KLOKMOSE

Associate Professor, University of Aarhus

Examineur

Titre: Substrats et instruments interactifs pour les historiques de l'interaction

Mots clés : médias temporels, historiques de l'interaction, interaction humain machine, interaction instrumentale, substrats interactifs, coordination

Résumé : Dans le monde numérique comme dans le monde physique, nos interactions avec les objets laissent des traces qui racontent l'histoire qui les a façonnés au fil du temps. Ces données historiques peuvent être consultées par les utilisateurs afin de mieux comprendre les étapes qui ont conduit à l'état actuel de leur système. Elles peuvent également être redocumentées afin d'arranger l'historique d'une manière plus compréhensible pour les utilisateurs. Dans des environnements collaboratifs, les utilisateurs peuvent être amenés à partager ces données, afin de mieux coordonner ou synchroniser leur travail d'équipe. Des travaux antérieurs ont tenté de démontrer les avantages des historiques partagés entre applications, mais les implémentations actuelles des historiques dans les systèmes interactifs continuent de confiner les historiques à leur application d'origine. Les utilisateurs ne peuvent pas croiser leurs historiques pour corréler les événements qui se sont produits dans différentes applications.

Dans cette thèse, je montre que concevoir des historiques de l'interaction pouvant être partagés entre les applications et les utilisateurs faciliterait la navigation, la compréhension et la réutilisation des données historiques. J'ancre le début de mes travaux dans le cas de l'écriture collaborative afin d'explorer des écologies de traces et des usages familiers, mais néanmoins complexes. J'identifie les pratiques récurrentes et les problèmes liés à l'utilisation des données historiques en interrogeant des utilisateurs habitués de l'écriture collaborative, et je mène plusieurs activités de conception basées sur les observations qui en découlent. Je décris ensuite un premier système en tant que preuve de concept intégrant deux outils résultant de ces activités de conception. Ce système intègre également la première itération d'une structure unique pour les données d'historique partagées entre applications et utilisateurs. Les résultats des études utilisateurs menées sur ce système montrent que ces derniers expriment effectivement le besoin de disposer d'historiques d'interaction unifiés et personnalisables.

En compilant les données recueillies au cours de ces activités de recherche et en me basant sur des travaux antérieurs concernant les "médias dynamiques partageables" et les substrats d'interaction, je décris un cadre permettant de concevoir des historiques d'interaction plus flexibles. Je présente Steps, une structure d'unification des données historiques qui intègre un noyau d'attributs descriptifs qui préserve l'intégrité d'une trace entre les applications, et des attributs contextuels extensibles qui permettent aux utilisateurs de modéliser leurs historiques en fonction de leurs besoins. Je présente ensuite OneTrace, un prototype basé sur les Steps. Son implémentation suit mon cadre descriptif pour les historiques inter-applications et définit l'historique comme un matériau numérique à façonner par l'utilisation d'outils dédiés. Je discute des opportunités offertes par cette approche pour réaliser un changement de paradigme sur la façon dont nous concevons les historiques et leurs outils.

Title: Interaction Substrates and Instruments for Interaction Histories

Keywords : temporal media, interaction history, human computer interaction, instrumental interaction, interactive substrates, coordination

Abstract : In the digital world, as in the physical world, our interactions with objects leave traces that tell the story of the actions that shaped these objects over time. This historical data can be accessed by end users to help them better understand the steps that led to the current state of their system. These traces can also be reused for activities such as re-documenting their own history to arrange it in a way that they find more understandable. Users may also be led to share these data in collaborative environments, to better coordinate and synchronize their work. While previous work has attempted to show the benefits of cross-application histories, current implementations of interaction histories in interactive systems tend to tie history data to their source application. This prevents users from cross-referencing historical data to review and correlate events that occurred in different applications.

In this thesis, I argue that designing interaction histories that can be shared among applications and users would support browsing, understanding and reusing historical data. I first ground my work in the use case of collaborative writing to explore relatable yet complex traces ecologies and interaction history use. I identify recurring practices and issues with the use of history data by interviewing knowledge workers and conducting several design activities based on these observations. I describe a first proof-of-concept system integrating two history instruments resulting from these design activities, and the first iteration of a unifying structure for historical data to be shared among applications and users. The results of user studies show that users indeed express a need for unified and customizable interaction histories.

Compiling the data gathered during these research activities and based on previous works about “Dynamic Shareable Media” and the Interaction Substrates and Instruments model, I describe a framework to help create more flexible interaction histories. The goal is to describe how to design interaction history systems that would help users take control of their historical data. I introduce Steps, a structure for unifying historical data that includes descriptive core attributes to preserve the integrity of a trace across applications, and extensible contextual attributes that let users reshape their histories to suit their needs. I then introduce OneTrace, a proof-of-concept prototype based on Steps that follows my descriptive framework for cross-application histories and defines interaction histories as digital material to be shaped by digital tool use. I discuss the opportunities offered by this approach to support a shift in paradigm on how we design and interact with interaction histories.

Synthèse du contenu du manuscrit

Dans cette thèse, je soutiens que la conception des historiques d'interaction devrait s'inspirer de théories telles que le modèle des substrats et instruments d'interaction, ou le concept de médias dynamiques partageables afin de mieux aider les utilisateurs à parcourir, comprendre et réutiliser leurs historiques. La création d'objets représentant les historiques, et d'instruments d'historique de l'interaction plus flexibles permettrait aux utilisateurs de croiser leurs données d'historique depuis n'importe quelle application et de les utiliser avec n'importe quel outil qu'ils trouveraient adapté à leur activité.

Je soutiens cette affirmation en identifiant et en abordant les problèmes spécifiques de la conception actuelle des historiques d'interaction, ce qui m'a conduit à mettre en œuvre un système d'enregistrement d'historique inter-applications et deux instruments d'historique d'interaction distincts. Je montre que la conception de tels systèmes interactifs aide les utilisateurs à manipuler les traces et à comprendre la chronologie des événements de leur environnement numérique.

Inspiré par des travaux connexes, l'objectif principal de cette thèse est de fournir un cadre qui aide à comprendre et à concevoir de tels environnements, et de soutenir un changement de paradigme dans la façon dont nous pouvons concevoir et interagir avec les historiques d'interaction.

Au cours de mes travaux de thèse, j'ai appliqué des méthodes empruntées à la fois au génie logiciel, à la psychologie et aux sciences sociales. Pour équilibrer mon approche multidisciplinaire, j'ai triangulé entre les perspectives théoriques, empiriques et conceptuelles de ces disciplines.

J'ai d'abord examiné les connaissances actuelles sur la manière dont l'utilisation des historiques d'interaction peut contribuer à améliorer le rappel d'événements passés, avec des travaux encourageant le partage d'historiques inter-applications et des travaux montrant les bénéfices de proposer différentes représentations d'un même historique. J'ai associé ces travaux antérieurs à des travaux théoriques sur la manière dont les individus se coordonnent au cours d'activités communes avec pour support leurs traces d'activités personnelles et partagées. En m'appuyant sur le concept d'écologie d'artefacts dans le contexte de l'écriture collaborative, j'explique comment la coordination peut être liée à la conception des historiques d'interaction et des outils disponibles aux différents membres de l'équipe.

J'ai utilisé la méthode des entretiens contextuels pour recueillir les premières informations empiriques sur les pratiques des utilisateurs et les ruptures avec les histoires, et j'ai utilisé des questionnaires pour compléter les données recueillies à partir des observations et des entretiens. J'ai utilisé la méthode de l'analyse thématique comme outil d'analyse qualitative des entretiens.

Les connaissances empiriques que j'ai pu recueillir m'ont permis de concevoir un prototype de système d'enregistrement des données d'historiques basé sur une structure de données unifiée que j'ai appelé "Steps". J'ai également utilisé des méthodes de brainstorming et de conception participative pour concevoir deux outils différents permettant d'interagir avec les historiques appelés "TraceLens" et "TracePicker". J'ai utilisé la théorie générative de l'interaction instrumentale comme outil de pour itérer sur la conception de ces deux prototypes. J'ai mené deux études d'observation sous forme d'entretiens et de scénarios. J'ai observé comment les participants utilisaient mes prototypes dans un contexte donné et j'ai évalué leur perception d'une telle approche par rapport à leurs habitudes.

A la suite de ces travaux de conception, j'ai pu repenser mon modèle d'historiques au travers du prisme théorique du modèle des substrats et instruments d'interaction et des travaux sur les logiciels malléables. J'ai ensuite décrit un modèle pour la conception d'environnements basés sur les substrats et les instruments d'historiques de l'interaction.

Basé sur la technologie "Webstrates" développée par l'université d'Aarhus, j'ai implémenté une version robuste de mon système basé sur mon modèle d'historique appelé "OneTrace", permettant aux utilisateurs d'accéder à un espace de travail qui peut être partagé. Dans cet espace de travail, les utilisateurs peuvent ouvrir leurs applications pour travailler, éditer, consulter du contenu, et donc générer de l'historique sous forme de "Steps". Ils peuvent ensuite consulter leur historique avec les instruments et représentations de leur choix en les important également dans l'espace de travail.

Certains des résultats et des figures de ce manuscrit ont été précédemment publiés dans l'article de revue suivant : "OneTrace : Improving Event Recall and Coordination With Cross-Application Interaction Histories" (doi: 10.1080/10447318.2024.2332848).

ALEXANDRE BATTUT

INTERACTION SUBSTRATES AND INSTRUMENTS FOR
INTERACTION HISTORIES

Substrats et Instruments Interactifs pour les Historiques de l'Interaction

PHD THESIS – UNIVERSITÉ PARIS-SACLAY

June 24, 2024

ABSTRACT

In the digital world, as in the physical world, our interactions with objects leave traces that tell the story of the actions that shaped these objects over time. This historical data can be accessed by end users to help them better understand the steps that led to the current state of their system. These traces can also be reused for activities such as re-documenting their own history to arrange it in a way that they find more understandable. Users may also be led to share these data in collaborative environments, to better coordinate and synchronize their work. While previous work has attempted to show the benefits of cross-application histories, current implementations of interaction histories in interactive systems tend to tie history data to their source application. This prevents users from cross-referencing historical data to review and correlate events that occurred in different applications. In this thesis, I argue that designing interaction histories that can be shared among applications and users would support browsing, understanding and reusing historical data. I first ground my work in the use case of collaborative writing to explore relatable yet complex traces ecologies and interaction history use. I identify recurring practices and issues with the use of history data by interviewing knowledge workers and conducting several design activities based on these observations. I describe a first proof-of-concept system integrating two history instruments resulting from these design activities, and the first iteration of a unifying structure for historical data to be shared among applications and users. The results of user studies show that users indeed express a need for unified and customizable interaction histories. Compiling the data gathered during these research activities and based on previous works about “Dynamic Shareable Media” and the Interaction Substrates and Instruments model, I describe a framework to help create more flexible interaction histories. The goal is to describe how to design interaction history systems that would help users take control of their historical data. I introduce Steps, a structure for unifying historical data that includes descriptive core attributes to preserve the integrity of a trace across applications, and extensible contextual attributes that let users reshape their histories to suit their needs. I then introduce OneTrace, a proof-of-concept prototype based on Steps that follows my descriptive framework for cross-application histories and defines interaction histories as digital material to be shaped by digital tool use. I discuss the opportunities offered by this approach to support a shift in paradigm on how we design and interact with interaction histories.

RÉSUMÉ

Dans le monde numérique comme dans le monde physique, nos interactions avec les objets laissent des traces qui racontent l'histoire qui les a façonnés au fil du temps. Ces données historiques peuvent être consultées par les utilisateurs afin de mieux comprendre les étapes qui ont conduit à l'état actuel de leur système. Elles peuvent également être re-documentées afin d'arranger l'historique d'une manière plus compréhensible pour les utilisateurs. Dans des environnements collaboratifs, les utilisateurs peuvent être amenés à partager ces données, afin de mieux coordonner ou synchroniser leur travail d'équipe. Des travaux antérieurs ont tenté de démontrer les avantages des historiques partagés entre applications, mais les implémentations actuelles des historiques dans les systèmes interactifs continuent de confiner les historiques à leur application d'origine. Les utilisateurs ne peuvent pas croiser leur historiques pour corréler les événements qui se sont produits dans différentes applications. Dans cette thèse, je montre que concevoir des historiques de l'interaction pouvant être partagés entre les applications et les utilisateurs faciliterait la navigation, la compréhension et la réutilisation des données historiques. J'ancre le début de mes travaux dans le cas de l'écriture collaborative afin d'explorer des écologies de traces et des usages familiers, mais néanmoins complexes. J'identifie les pratiques récurrentes et les problèmes liés à l'utilisation des données historiques en interrogeant des utilisateurs habitués de l'écriture collaborative, et je mène plusieurs activités de conception basées sur les observations qui en découlent. Je décris ensuite un premier système en tant que preuve de concept intégrant deux outils résultant de ces activités de conception. Ce système intègre également la première itération d'une structure unique pour les données d'historique partagées entre applications et utilisateurs. Les résultats des études utilisateurs menées sur ce système montrent que ces derniers expriment effectivement le besoin de disposer d'historiques d'interaction unifiés et personnalisables. En compilant les données recueillies au cours de ces activités de recherche et en me basant sur des travaux antérieurs concernant les "médias dynamiques partageables" et les substrats d'interaction, je décris un cadre permettant de concevoir des historiques d'interaction plus flexibles. Je présente Steps, une structure d'unification des données historiques qui intègre un noyau d'attributs descriptifs qui préserve l'intégrité d'une trace entre les applications, et des attributs contextuels extensibles qui permettent aux utilisateurs de modéliser leurs historiques en fonction de leurs besoins. Je présente ensuite OneTrace, un prototype basé sur les Steps. Son implémentation suit mon cadre de-

scriptif pour les historiques inter-applications et définit l'historique comme un matériau numérique à façonner par l'utilisation d'outils dédiés. Je discute des opportunités offertes par cette approche pour réaliser un changement de paradigme sur la façon dont nous concevons les historiques et leurs outils.

ACKNOWLEDGMENTS

Science is not a path one can walk alone. No one can take any credit for his work without acknowledging all the people who influenced them and their work by any means. I first want to thank my thesis jury, Myriam Lewkowicz, Tom Gross, Petra Isenberg and Clemens Klokrose for taking the time to review my work, and for providing me with insightful feedback and questions about my research.

I also want to express my gratitude towards my thesis supervisor Michel Beaudouin-Lafon, and to Wendy Mackay, who both in their own way made me believe in what I did, always encouraged me in my research and supported me during this complicated journey. It is almost frustrating to see how much I've learned with you both, when I may never be able to give back what you've given me.

I want to thank all the researchers that I've met at ex-Situ. First Viktor Gustafsson, Miguel Renom, Han Han, Elizabeth Walton and Jean-Philippe Rivière, for their warm welcome in the team when I just arrived, for their enthusiasm and their valuable tips and tricks on how to make good research and how to survive the PhD. Special thanks to Miguel Renom who shared the pain of the thematic analysis with me during lockdown.

Then I want to thank all the PhD students I met during my stay in ex-Situ : Capucine Nghiem, Tove Grimstad Bang, Martin Tricaud, Camille Gobert, Anna Offenwanger and Wissal Sahel. We all shared the difficulties, the pain, but also the joys of these first years of our academic career. I am grateful to be part of this generation of researchers, and I hope we will all find our place in this world of science.

I want to do a special thank you to a few people who had my back at any time: Arthur Fages, my work roommate, with whom we shared laughs, tears and beers, and I hope that we did not see the last of this friendship. I want to thank Alexandre Kabil, for his everyday support, his everlasting kindness at all times. Thank you Romane Dubus, for sharing rants together, about life and about research, but in the end, we always ended up motivating each other. Thank you Junhang Yu for your friendship and unflinching good mood. I really hope you found a place where you could use your undeniable talent as a designer.

A last round of special thanks to Nicolas Taffin, a great web designer/editor/graphic designer/everything you may think of. I will always remember working with you and Wendy on VideoClipper, you shared so much knowledge with me and I really hope our path will cross again.

Finally thank you Clemens Klokmoose and all the Webstrates team for their support, their expertise and for sharing their knowledge. I am far from being the best in terms of technical knowledge and yet, maybe without knowing you encouraged me with your technology to step out of my comfort zone and have weird crazy ideas without worrying too much about if I would be able to prototype it or not.

Je voudrais remercier mes parents Jean-Philippe et Véronique, pour leur support inflexible tout au long de ces années compliquées. Je remercie également mon petit frère Benjamin, avec qui je suis heureux de partager un humour raffiné et sensible, et avec qui échanger un message par mois me remonte toujours le moral. Je remercie également mes amis de Toulouse, dont la liste est longue mais qui ont toujours su être présents malgré ma localisation lointaine, vérifiant par la même occasion l'adage "loin des yeux, près du cœur".

Bien sûr merci à toute l'équipe du laboratoire d'informatique interactive de l'Enac pour leur accueil, leur patience et leur support, et avec qui il me tarde de continuer à travailler. Merci également aux membres de l'équipe ICS de Toulouse pour m'avoir initié à la recherche en IHM et m'avoir donné envie d'emprunter cette voie.

Last but not least, I wanna thank Wafa. I wanna thank you for always believing in me. I wanna thank you for supporting me in doing all this hard work. And I wanna thank you for always being you at all times.

This work was partially supported by European Research Council (ERC) grant n°695464 "ONE: Unified Principles of Interaction".

CONTENTS

1	INTRODUCTION	1
1.1	Thesis statement	3
1.2	Methods	3
1.3	Thesis Overview	4
2	BACKGROUND AND CONTEXT	7
2.1	Interaction Histories	8
2.1.1	What is an Interaction History?	8
2.1.2	Interaction history uses	10
2.1.3	Supporting Interaction Histories context recall	12
2.2	Instrumental Interaction and Interaction Substrates	18
3	UNDERSTANDING INTERACTION HISTORY USE	25
3.1	Setting the scene: why collaborative writing?	26
3.1.1	Coordination	26
3.1.2	Territoriality	27
3.1.3	Ecologies	27
3.1.4	The Collaborative Writing Traces Ecology	29
3.2	Study 1 : Interviews with Knowledge Workers	31
3.2.1	Method	31
3.2.2	Results : Thematic Analysis	32
3.3	Additional Insights with Questionnaires	37
3.3.1	Method	37
3.3.2	Results	38
3.4	Conclusion and Discussion	40
4	TOWARDS CROSS-APPLICATION INTERACTION HISTORY SYSTEMS	43
4.1	Steps : A Unified Data Structure for Interaction Histories	44
4.2	The OneTrace system : a cross-application interaction history logging system	46
4.3	Conclusion	48
5	EXPERIMENTING WITH CROSS-APPLICATION HISTORY INSTRUMENTS	49
5.1	Design Goals and Research Question	50
5.2	TraceLens : An instrument to navigate cross-application histories	51
5.3	Study 2 : Assessing TraceLens benefits	55
5.4	Results and discussion	59
5.5	TracePicker : An instrument to enhance history data contextualization	61
5.6	Study 3 : Assessing TracePicker benefits	65
5.7	Results and Discussion	67
5.8	Conclusion	70

6	INTERACTION HISTORY SUBSTRATES	73
6.1	Understanding history artifacts : what is a timeline ?	76
6.2	From Steps to history substrates : iteration on the Steps model	79
6.3	A global framework for interaction history substrates and instruments.	82
6.4	Iterating on the Proof-of-Concept : OneTrace and Webstrates	84
6.4.1	Why is Webstrates Relevant?	84
6.4.2	Implementing OneTrace with Webstrates.	86
6.5	Conclusion and Discussion	89
7	CONCLUSION	93
7.1	Overview of the contributions	93
7.2	Limitations and future work	96
	BIBLIOGRAPHY	99
A	APPENDIX	111
A.1	Privacy Concerns and Shared Histories Questionnaires	111
A.2	TraceLens Study Questionnaires	114
A.3	TracePicker Study Questionnaires	117

LIST OF FIGURES

Figure 1	The different interaction history representations.	9
Figure 2	Switch! user interface.	10
Figure 3	The Chronicle interaction histories.	11
Figure 4	The CAAD's activity display.	12
Figure 5	The Footprints scroll bar.	13
Figure 6	LifeFlow's user interface.	14
Figure 7	YouPivot's enhanced history.	15
Figure 8	Time Casting system.	16
Figure 9	Leavo's activity overview.	17
Figure 10	The Xerox Star User Interface.	19
Figure 11	A rich substrate and its instruments.	20
Figure 12	The Passages environment.	21
Figure 13	The Vistrates' architecture.	22
Figure 14	Artifact ecologies in collaborative writing.	28
Figure 15	Collaborative writing applications and traces ecologies.	30
Figure 16	Examples of drawings created by interviewees.	31
Figure 17	Iterations on the thematic analysis.	32
Figure 18	Willingness to share tools and histories (n=27).	38
Figure 19	Interest in cross-application history tools (n=27).	39
Figure 20	The three main components of a Step.	44
Figure 21	The OneTrace desktop User Interface.	46
Figure 22	Brainstorming results examples.	52
Figure 23	The TraceLens concept.	54
Figure 24	The TraceLens prototype implementation.	55
Figure 25	TraceLens questionnaires results.	58
Figure 26	Participatory design workshop results examples.	62
Figure 27	The TracePicker concept.	63
Figure 28	The TracePicker prototype implementation.	64
Figure 29	The PickLines provided for the second task of the study.	66
Figure 30	TracePicker questionnaires results.	68
Figure 31	The StickyLines instrument.	76
Figure 32	History instruments and how they interact on substrates to create a new substrate.	77
Figure 33	Iteration on the Steps model.	80
Figure 34	A framework for writing Steps.	82
Figure 35	A framework for reading and interacting with Steps.	83

Figure 36	OneTrace implementation with Webstrates : Architecture and Life Cycle.	86
Figure 37	OneTrace implementation with Webstrates : User Interface.	88

LIST OF TABLES

Table 1	Overview of the five identified themes.	33
Table 2	Instrumental interaction applied to TraceLens and StepLine.	53
Table 3	Questionnaire items related to the TraceLens study questions.	57
Table 4	Questionnaire items related to the TracePicker study questions.	67

INTRODUCTION

In the digital world, as in the physical world, our interactions with our environment and objects leave traces of interactions [71, 78]. These pieces of history tell the story of the actions that shaped an object over time, attesting its wear [4, 35], that result in the object's current state. In interactive systems, the use of tools and commands log history data. These data can be accessed as raw log files to analyze users' activities, whether it be for supporting them through live recommendations or for later investigations to improve features and tools [14, 22]. But they can also be accessed as interactive digital objects. If the software editors identified it as a useful feature for end users, these traces of interaction are typically sorted and found in panels and layouts such as timelines or ordered lists of elements. They can be sorted temporally by laying out events according to the time when they occurred or spatially by linking a trace to the place in the document where the interaction occurred.

When these traces can be accessed by end users, it helps them better understand the steps that led to the current state of their system [29]. These traces can also be reused for activities such as re-documenting their own history to arrange it in a way that they find more understandable [90]. Current implementations of interaction histories in interactive systems tend to tie history data to their source application. Interaction histories are then to be displayed and interacted with in dedicated tools only accessible from that source application. For example, a document's modification history is displayed in the text editor's history [35] while a web browsing history is displayed in the browser's history [31].

While each type of history can describe precisely the actions performed on the object and the time frame of these interactions, it is almost never possible to browse them in other applications, or enriched with other applications' histories. In the course of an activity, users may interact with several tools and digital objects in parallel, which will fragment their history and leave traces in different places. This fragmentation makes it difficult for users to reconstruct and recall the global context in which a past event occurred and that needs to be re-contextualized [73, 74]. Indeed, if time and space can be good ways to recall past events, it is also often by associating past actions with one another that individuals remember the details of history elements. It is then difficult for users to review and correlate events that

occurred in different applications, as they have no way to link the different histories together.

Today's interface and any associated tools and features are mainly based on the WIMP (windows, icon, menu, pointing) interaction model, and the desktop metaphor that is often associated with it, that was introduced in the early eighties [81]. Since then, our desktops have evolved into constellations of applications and documents that offer little to no interoperability among each other. On the other hand, researchers have explored and described new post-WIMP interaction theories and principles, conceptual models for designing new interfaces supporting modern interactions. These led to generative principles [12] that support the design of more flexible and extensible interactive systems. Previous work on this matter also include a new interaction model, the interaction substrates and instruments model [11], that supports the design of digital objects and instruments that are polymorphic and reusable, and that operationalize the notion of affordances. The concept of "Shareable Dynamic Media" [43] also showed the feasibility of distributed, shareable and malleable software.

The shortcomings of current interaction history tools and how they fail to support users' actual practices with histories may come from the way we design interaction histories in the first place. Even when trying to support history data sharing across applications, previous work has led to designs that still tie histories to a single feature to be used in a single specific application. And thus, several questions remain : How to support users in easily collecting histories from different sources? How to let them browse their histories with the instruments they prefer, independently of their source application? How to let them customize their histories in a way that makes sense to them?

1.1 THESIS STATEMENT

In this thesis, I argue that the design of interaction histories should take inspiration from theories such as the interaction substrates and instruments model or the concept of shareable dynamic media to better support users in browsing, understanding and reusing their histories. Creating more flexible interaction history objects and instruments would let users cross-reference history data from any application and use it with any tool they may find suitable for their activity. I support this statement by identifying and addressing specific issues of today's interaction histories design, leading me to implement a cross-application history logging system and two distinct interaction history instruments. I show that designing such interactive systems helps users manipulating traces and understanding the timeline of events of their digital environment. Inspired by related work, the main goal of this thesis is to provide a framework that helps understanding and designing such environments, and support a shift in paradigm in how we can design and interact with interaction histories.

1.2 METHODS

During this thesis, I applied methods borrowed from both software engineering, psychology and social sciences. To balance my multidisciplinary approach, I triangulated between the theoretical, observational and design perspectives of these disciplines [54].

I first looked at the current knowledge on how the use of interaction histories can help enhance recall of past events. I associate this previous work to theoretical work on how individuals coordinate with each other during joint activities [56]. Through the lens of the artifact ecologies concept [48], I explain how coordination can be linked to interaction histories design. I used the method of story interviews [52] to gather first empirical insights about users' practices and breakdowns with histories, and employed questionnaires to complement the data gathered from observations and interviews. I used the method of thematic analysis [19] as a tool for the qualitative analysis of the interviews.

The empirical knowledge I gathered allowed me to design a prototype system for logging history data based on a unified data structure. I also used brainstorming and participatory design methods to design two different tools to interact with histories. I used the generative theory of instrumental interaction [12] as a tool to iterate on these two designs. I conducted two observational studies in the form of two interview- and scenario-based studies. I observed how partici-

pants used my tools in a given context and assessed their perception of such an approach compared to their habits.

Finally, I reflected on my work through the theoretical lens of the interaction substrates and instruments model [10] and works on malleable software [43]. I then described a framework for designing environments based on interaction history substrates and instruments.

1.3 THESIS OVERVIEW

The outline of this manuscript articulates the different contributions and works produced during this thesis to present a coherent path that led me to my thesis statement.

CHAPTER 2 defines the framing of my research on interaction histories. It reviews related work on interaction histories, how users interact with them and what improvements still need to be done. It shows to what extent improving interaction history usage across users' workspaces would improve their understanding of a digital object's timeline and past events' context recall. I also review related work on the concepts of instrumental interaction, interaction substrates and malleable software which are an important part of the reflection about interaction histories in this thesis.

CHAPTER 3 starts by reviewing related work about interaction history usage in collaborative text editing settings, and defines a grounded use case to base future research on. It then describes the empirical contribution of this thesis with the interview of twelve knowledge workers supplemented by a twenty-five respondents questionnaire, both about interaction history usage in collaborative writing settings. The results of this study are analyzed using thematic analysis that allows me to define the main practices and issues with interaction histories described by the participants.

CHAPTER 4 starts with gathering insights from the two first chapters to describe a unified structure for history data which I called "Step". This data structure is a way for me to both introduce a new approach for designing interaction histories and implement a cross-application history system. I then show how I implemented OneTrace, a proof-of-concept system based on Steps. OneTrace logs the actions performed by users and their context as Steps, and shares them across applications and users.

CHAPTER 5 describes two different projects that aimed at experimenting with designing instruments for OneTrace. Both projects aimed at implementing an instrument that answered a design goal extracted from the interviews. Both projects started with participatory

design activities. Then both designs were improved by reflecting on these using the instrumental interaction generative theory. Both instruments were implemented in OneTrace and helped to gather insights on users' perception of such cross-application history tools during two interview- and scenario-based user studies.

CHAPTER 6 presents a framework for designing cross-application history environments based on the substrates and instruments interaction model. It is made by reflecting on my first assumptions on a unified history data structure, and my observations from the studies I conducted with OneTrace. It aims at giving a better understanding of what it is to interact with graphical objects made of interaction history elements, and provides insights for the design of better history instruments. As a proof-of-concept, I describe the implementation of an updated version of OneTrace based on this framework.

CHAPTER 7 concludes with a review of the work described in this manuscript, and the contributions it makes. Then I discuss the main limitations of this thesis, and give directions for future work.

Some of the results and figures in this manuscript were previously published in the following journal article: *OneTrace: Improving Event Recall and Coordination With Cross-Application Interaction Histories* [8].

BACKGROUND AND CONTEXT

In this thesis, my first interest is to study interaction histories as first class objects. In this chapter, I first describe the underlying concept behind the term “interaction histories” that can be used to describe different types of data or representations, and I specify what I mean by interaction history in this manuscript.

I then explain the different ways interaction histories can be used, and differentiate the use of histories for automating tasks and for data analysis purposes from the use of data by users themselves in order to organize and understand past activities. I elaborate on the latter and explain how users may take advantage of the traces left on digital objects. To this end, I conduct background research on related work investigating how users contextualize past events in interaction histories. Previous studies show a need for a more flexible approach in terms of design for interaction histories, and I argue that while several studies point out the benefits of this flexibility to better understand past activities on users’ side, they do not fully embrace it in terms of design.

I acknowledge that my point of view is largely inspired by previous work on interaction substrates and instruments [9, 11] and the concept of malleable software [43]. I present both of these concepts and review related work to show the possibilities offered by flexible and extensible interactive systems. This background research leads to first conclusions about interaction history tools and how to improve them.

2.1 INTERACTION HISTORIES

In this section, I define interaction histories, describe them as interactive systems and review related work on how previous research tried to enhance the use of such systems. This leads to my first motivation: enhancing interaction history tools by making interaction histories more flexible.

2.1.1 *What is an Interaction History?*

Interaction history as an object of study is not well defined in the literature. To make an attempt at defining it I will first clarify what history in general means. Among the multiple definitions proposed in the different available dictionaries, two of them stand out. The first one, from the Cambridge dictionary [1], defines history as "*(a record of) past events considered together, especially events of a particular period, country, or subjects*". This definition puts the emphasis on the fact that history is a way to organize past events that are related with each other depending on a main factor, which can be a temporal factor but can also be centered around a place or an individual. The second one, from the Oxford dictionary [2], defines history as "*the past events connected with the development of a particular place, subject, etc.*". With this definition we have the additional idea of "development". It means that having access to the history of an object of study gives the possibility to understand the current state of this object. From these definitions I describe a general history as follows :

"a history is a succession of events considered together and centered around the same object of study that gives key elements to understand the current state of this object of study."

In this thesis, I use the term interaction history to describe the events that occurred in the past and that are logged by the system when users interacted with their digital environment. I may sometimes refer to interaction history as "traces". Indeed, as in the physical world, our interactions with objects leave traces as interaction history [71, 78]. Based on the above definition of history, an interaction history can be defined as follows :

"an interaction history is a succession of past user's events recorded by the system they interacted with."

This definition deliberately does not include the way it can be presented to users or the way it can be reused or analyzed further. Indeed, in the literature, even though my definition is generic enough to encompass the different systems described as interaction histories, the digital objects representing interaction histories can take various

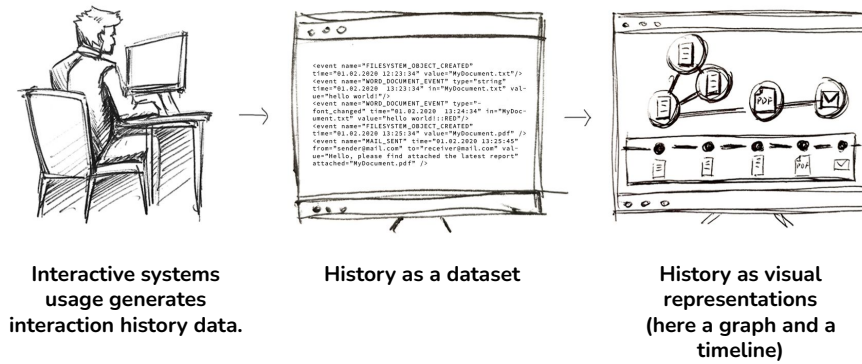


Figure 1: Different interaction history representations, inspired by Schmidt et al. [74]. Users interact with the system, generating a dataset of user events logged by the interactive system, and visual representations can be built thanks to these data (here with a timeline and a graph that interprets the interaction history logged by the system).

forms. These forms include different levels of interactivity and accessibility on the user’s side.

These differences can be classified in different levels of representation. For example, works such as Schmidt’s et al. [74, 75, 77] studies on interaction histories data management, analysis and visualizations help us identify these different layers of interaction histories (Figure 1). First, interaction history is a dataset of elements that provide information about what operations the user performed and on which objects. Each of these elements has a temporal component about the interaction’s occurrence, and describes the past interaction with a granularity depending on the application logging the event. This kind of representation is not usually meant to be accessed by the user. But information from these datasets can be aggregated or interpreted by interactive systems to provide users with a more relevant representation. This second layer of representation can consist of graphs or timelines, layouts based on the temporal component of histories, representing history as time series [3, 58, 85], sometimes combining different ones to complete each other [29]. It can also adopt non-temporal layouts but more spatial representations [4, 35] if the temporal component is not the main attribute of interest to the user. Generally speaking, it is a visual representation of the dataset aimed at being understandable and usable by the end user.

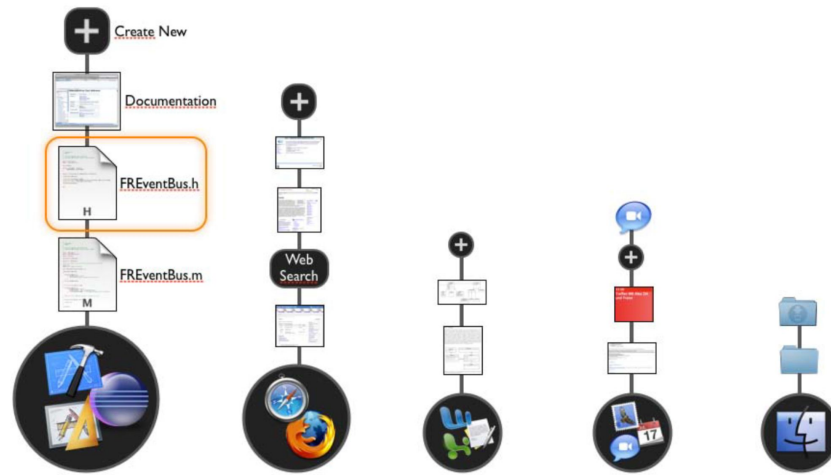


Figure 2: Maalej's et al. [51] Switch! user interface showing recommendations based on switching history.

2.1.2 Interaction history uses

Now that I defined what is an interaction history, and what kind of digital objects I am referring to with this concept, I want to describe the different uses of interaction histories that I identified in the literature, and characterize which one this thesis will be focusing on.

First, there are systems that process interaction history data in order to better support the user in their tasks, and to adapt the system to their needs. Such systems use task detection or context detection models to understand the user's current activity based on their interaction history and their current inputs, such as Rath's et al. [67] UICO system, Shen's et al. [79] Task Tracer system, or Lokaicznyk's et al. [50] context mapping algorithm. Such systems are built to proactively provide recommendations to the user, thanks to additional interface features, such as Maalej's et al. [51] Switch! system that helps users when switching applications in a given work context and thus improving their efficiency in switching applications on their system (Figure 2).

Processing interaction history datasets is not only useful for supporting features and systems providing recommendations. When knowledge work becomes an object of study, as with any other object of study, interaction history analysis becomes a powerful tool to better understand users' activities and behavior on their interactive systems, as exemplified by the works of Belin et al. [14] or Carillo et al. [22]. In that case too, task mining can be used to enrich histories such as Parnin's et al. [66] revision history system, to refine the analysis of histories and to make the information more complete. Overall, this first category of use can be defined as an a posteriori analysis of interac-

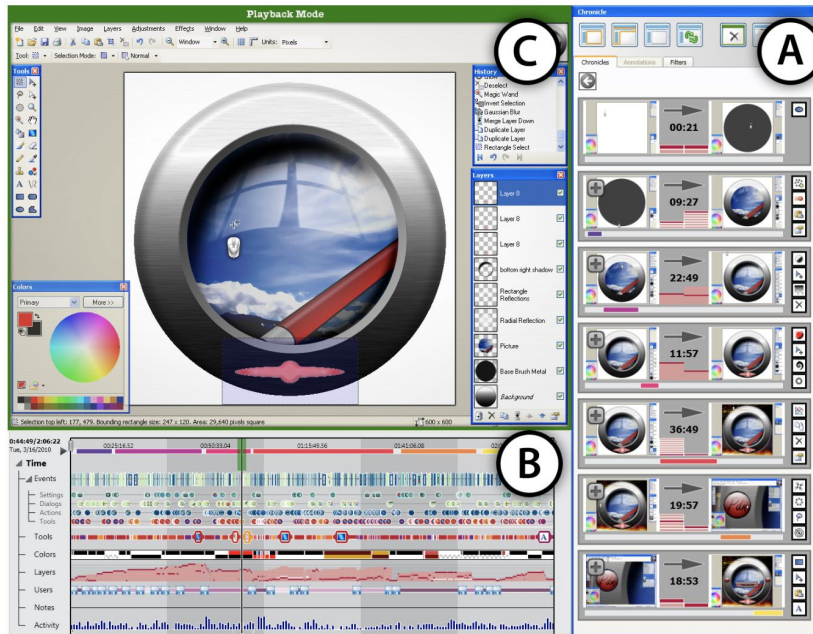


Figure 3: Grossman’s et al. [29] Chronicle system with three ways to browse interaction history : (a) The main Chronicle window; (b) The timeline; (c) The application/playback window.

tion history datasets, whether it be for live support of users’ activities or for later investigation of users’ behavior.

Second, we have systems that let users browse and manipulate history data themselves. These systems support the user in understanding the timeline of a digital project and the events that led to its current state on different levels. First, they improve the way users navigate history and browse the available information. For example, Schmidt et al. [74] or Grossman et al. [29] (Figure 3) explore the different ways in which to visualize histories depending on which information users want to retrieve. Similarly, Freeman and Gelernter’s Lifestream system [25] and Rekimoto’s [69] Time-machine computing system organize the users’ working environment around their interaction history to better facilitate its use. These works underline the importance of interaction history tools and graphical representations that are designed around users’ practices to make them better grasp the general timeline of their work context. As explained by Schmidt et al. [76], a better understanding of interaction history as a whole provides a better awareness of the current state of the object of interest and helps plan future actions.

Manipulating history data is not just about browsing it. While in many interactive system, interaction histories support browsing by filtering time or keywords, users also remember events by their relationships to each other [74]. That type of use is often supported

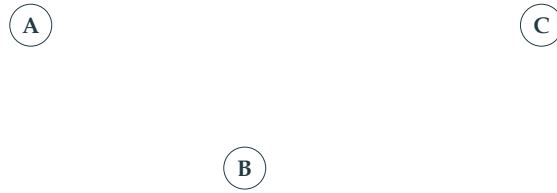


Figure 4: Caad's activity display [68]: (a) The left-most image was automatically generated by CAAD; (b) The second image illustrates a user-edited display; (c) The third image shows one group in more detail.

by allowing people to organize their interaction histories into collections of past events, creating a new digital object that is meaningful to the user. Systems such as Kaptelini's et al. [39] UEMA system and Morteo's et al. [57] Sphere Juggler support such history data organization as custom collections. Rattenbury et al. [68] go even further with their CAAD system by generating collections of events automatically, which the user can then tweak and combine to create a single meaningful object (Figure 4). This kind of appropriation and organization of past events to create new custom digital objects is reminiscent of personal information management, as described by Malone [55]. As people organize their workspace in a way that is meaningful to them and to better retrieve important documents, they do the same to retrieve information related to past events.

Across these different ways to interact with history data, whether by browsing or by manipulating past events, the users' goal is always to better understand the timeline of events that led to the current state of a digital object. I define this type of use as using history as a digital *instrument* that empowers users in managing and understanding past events rather than a simple dataset to analyze. This is the type of interaction history use that I will focus on in this thesis.

2.1.3 Supporting Interaction Histories context recall

As explained earlier, while users can retrieve information by filtering the interaction history by time or keywords, they also find past events by their relationships to other events. Previous work from the literature has tried to enhance interaction histories with features such as adding contextual cues to history data or giving information about

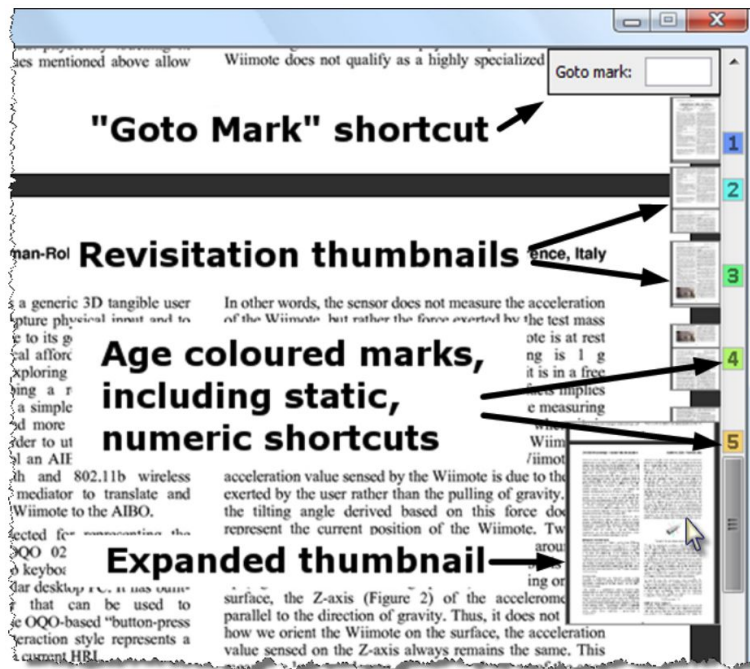


Figure 5: The Footprints scroll bar [4].

the temporal context in which a past event occurred to support users in their information retrieval activities.

For example, previous works have explored how to enhance past event recall by helping users reconstruct, as Rule et al. [73] describe it, the mental context of a past situation. To do so, interaction histories can be enhanced with visual facilitators. With ReadWear and EditWear, Hill et al. [35] present a new way to approach history data by displaying it as "*wear*" on the scroll bar of a document. This wear has varying intensity depending on the frequency of history data in a document's region. It was later revisited by Alexander et al. [4] (Figure 5) and their Footprint Scrollbar by adding marks and thumbnails on the wear. These visual cues provide an overview of the area impacted by past events, supported by a task cueing interface and the recency information communicated by the original "*wear*" from Hill et al. [35]. They allow for a more efficient revisitation of the document by better understanding the timeline of the document.

SenseMap's "history map" from Nguyen et al. [60] also explores this type of design with visual thumbnails. In this system, user's browsing history is displayed as a tree. To help users recall what they were doing, screenshots of the system in the form of thumbnails enrich this representation. The same approach is taken by Won et al. [88] and their Contextual Web History, a web browsing history enriched with thumbnails for each of the previously visited website, and by Hu's et al. [36] ScreenTrack that provides the user with a playback composed of screenshots of previously visited websites.

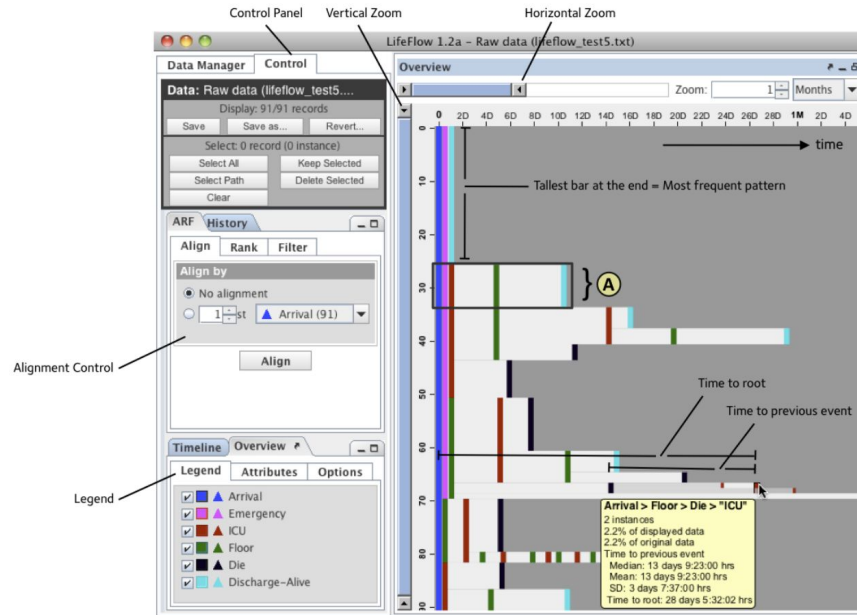


Figure 6: LifeFlow’s user interface [89] showing different samples of patients’ transfer histories.

While these works focus on triggering memories with visual cues, other works focus on the fact that, as Weick et al. [87] explain it, users may interpret past event as ensembles of actions rather than isolated actions. Translated into design goals for interaction history tools, it means that clustering interaction history data together is important to help users reconstruct context. This type of systems includes the UEMA, Sphere Juggler and CAAD systems [39, 57, 68] that I already mentioned. It is also interesting to add that the support that CAAD [68] offers for creating collections of history data is provided by task analysis, therefore building contextual structures representing the high-level task in which the documents were used. In Variolite’s and Verdant’s interactive histories [41, 42] of code editors, history data are visually connected together if the logged actions share the same goal. Their goal is to improve the use of version control in an attempt to mirror the user’s mental map and to help users reconstruct it.

Grouping traces by relationships has also been explored by Schmidt et al. [74] who found that linear representations are not suitable for finding context. They instead introduced graph representations to link traces together, for example based on the frequency of users switching between artifacts, in addition of the temporal structure offered by traditional timelines. In another context, Wongsuphasawat’s et al. [89] LifeFlow informs users about sequences of related events about hospital patients’ transfers and transportation instead of pro-

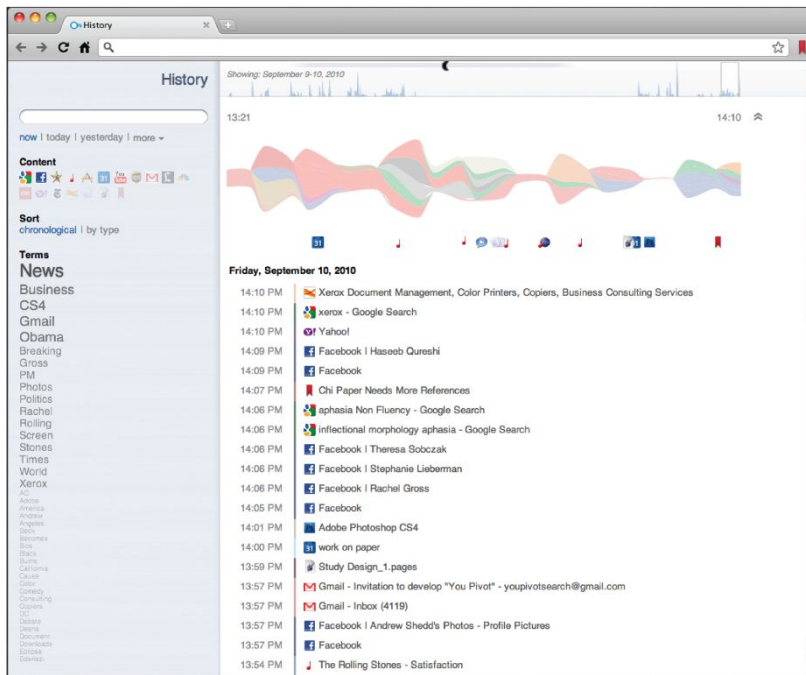


Figure 7: YouPivot's enhanced history [31].

viding a linear timeline (Figure 6). It gives a scalable visual overview of all the events that have happened around a patient once he or she has been taken care of, as sequences of events that can be summarized. The results show an improvement in understanding the timeline of events, and in detecting possible anomalies in their processes. Seeking to turn interaction history into a learning tool, Grossman et al. [29] developed Chronicle (Figure 3), a system that presents interaction history in three different ways so as to better communicate the information to the user by focusing on different types of information. In Chronicle's main window, history data are grouped into *chronicles*, sequences of actions represented as a single higher-level transition between two important states of the system. The two other windows are composed of a detailed timeline of commands that occurred on the document, and a playback of the document's life cycle. Therefore, this system provides three ways to communicate, browse, and understand a specific process that we want to teach, at different levels of granularity and using different media. Overall, all these works share the common goal of enriching history data with another layer of information to give it more meaning, either for navigational purposes by helping users focus on important parts of the history or for a better understanding of the timeline.

Reconstructing context to enhance recall has been explored by focusing on the users' activities while performing a task on the object of interest. This often requires referring to histories from other applications or tools that should be accessed together, or observing

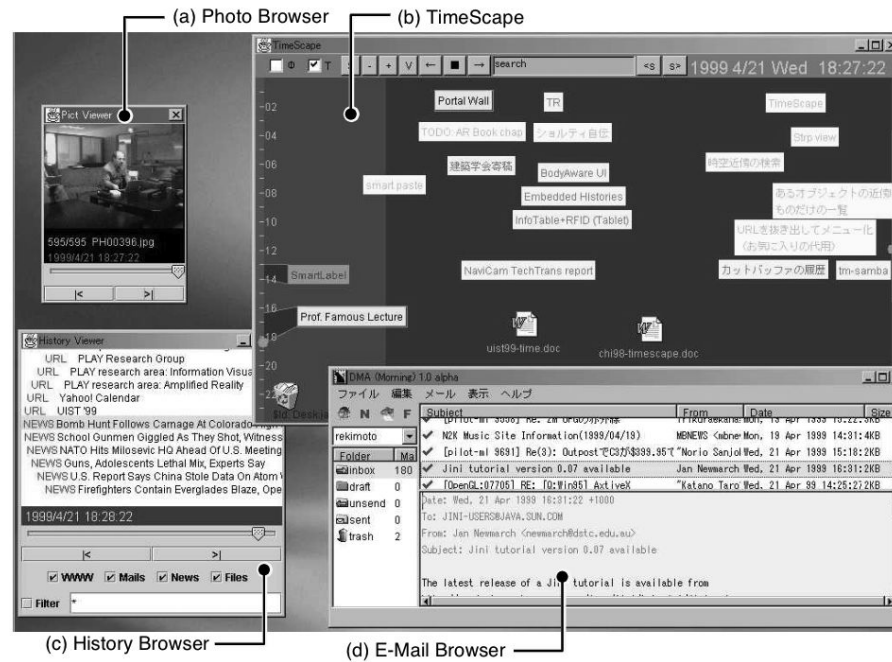


Figure 8: Time Casting [69] links multiple applications by exchanging time information.

the history of interactions at a higher level to understand the context of past actions, as explored by Nancel and Cockburn [59] with their Causality Model of Interaction History. The Switch! system [51] that I mentioned before supports users switching between different applications by remembering the context of the transitions between apps. This history-supported system observes interactions across the entire system and combines them with those performed on the object of interest to support the main activity. In line with this idea, Hailpern's et al. [31] YouPivot (Figure 7) helps users recall past activities by introducing contextual search. Time marks log user actions in any application within a specific time frame, which helps with contextual recall by associating their main activity with unrelated events, e.g. if the user remembers playing a specific song while modifying a document.

Waldner's et al. [84] WindowTrails also explores enriching histories with interactions from outside the application of interest. Acknowledging the fact that knowledge workers go back and forth between multiple application to seek, read, extract and compile information, it connects browsing histories and operation histories from individual applications to support information re-discovery, a cumbersome activity for this group of users. The CAAD system [68] we saw earlier takes a slightly different approach to regroup previously accessed documents into contextual structures, with a history system that is application-independent. The FileBiography system by

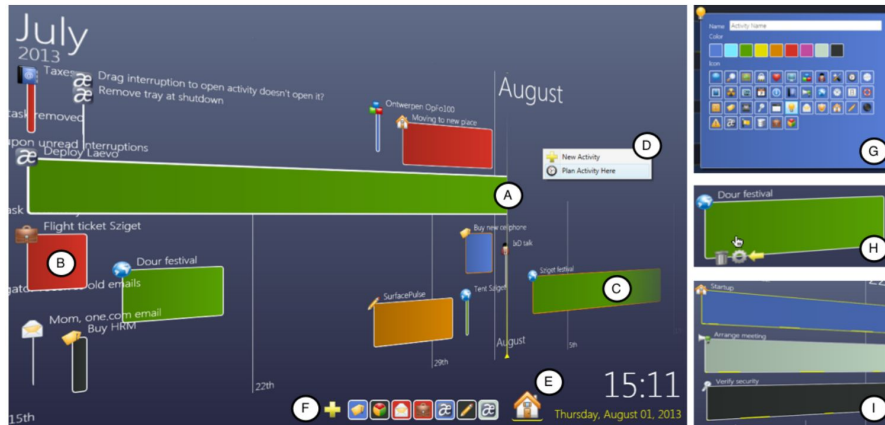


Figure 5. The activity overview through which activities are accessed and managed. It displays open ongoing activities (A), archived activities (B), planned activities (C), a context menu through which new activities can be created (D), the home activity (E) and to-do list (F). The three detailed images show the popup menu to edit activities (G), action buttons when hovering over an activity (H) and attention span lines (I).

Figure 9: Leavo's activity overview [37].

Lindley et al. [49] also supports cross-application history but uses a content-centered history that synthesizes activities performed on and with a file (modification, sharing, annotations and comments). By underlining relationships amongst history data, users can have a better understanding of a file's life cycle and its propagation among individuals.

Several projects explored a more application-agnostic approach for designing interactive systems that are based on timelines and temporality in general. With Lifestreams [25], Freeman et al. show an alternative to traditional file systems interfaces that gathers files into a single interactive timeline. The goal is to enhance personal information management by organizing files and past operations according to their chronology. With the TimeScape system, Rekimoto [69] presents an alternative to the current desktop environment in the form of a timeline organizing every digital content created on the computer. With TimeCasting (Figure 8), Rekimoto [69] builds on TimeScape and lets the different applications used in the system share temporal information, thereby linking the user's traces together. TimeCasting's design goal is to help users recall their history by associating the different activities they had at a given period.

In the same spirit, Jeuris et al. [37] developed the Leavo system which they define as a "temporal activity-centric desktop interface" (Figure 9). It offers an alternative to the usual desktop environment by organizing files and applications on a timeline depending on the activities performed, or the tasks to do in the future. While it is not only an interaction history system but also a planification system, it is important to note that supporting knowledge work is also about managing the different applications in the users' environment and

grouping them into workspaces dedicated to specific activities in order to be more efficient.

In summary, to enhance past event recall with interaction histories, the main answer from previous work has been to help users reconstruct the mental context of past activities, whether it be with visual helpers or collections of related events. These works show how histories from different places, applications and object are entangled with each other. However, most of these studies have focused on specific and closed environments [31, 84]. While they enhance sense-making for specific activities, they are focused on the initial purpose of the tool, which could undermine their benefits. Work environments are flexible and composed of numerous artifacts [37, 48] but these interaction history systems lack the flexibility to let users access new histories within their current work environment or re-use these tools for other activities. The works that inspired me the most are those that explore this flexibility through cross-application sharing of interaction histories such as Rekimoto's Time Machine Computing [69]. This aspect of interaction history tools design is the one that I want to investigate in more depth.

2.2 INSTRUMENTAL INTERACTION AND INTERACTION SUBSTRATES

My interest in making interaction histories more flexible is not only based on related work on interaction histories, but is also inspired by research on instrumental interaction, interaction substrates and malleable software. In this section I introduce these concepts which are part of the core of my thesis.

In the early eighties, Xerox introduced the Star User Interface [81, 82] (Figure 10). Along with Shneiderman's description of the principles of direct manipulation [80], it introduced the now most common interaction model known as "*Windows, Icons, Menus, Pointing*" (WIMP). WIMP-based user interfaces were initially designed for office workers, and introduced the desktop metaphor. While computers today can allow more interactions styles and modalities, from gesture-based interactions to tangible interaction, the dominant model for designing user interfaces remains the WIMP model. Current interactive systems are mainly organized as follows : our tools are applications, they allow us to interact with digital objects with a predefined set of commands, and the resulting object is often stored in a proprietary format. The content is not distributed among applications, inducing a lack of interoperability and confining the user to the commands that are available in their application. Current interfaces lack flexibility as users cannot configure their tools to change their behavior as they need it, the supported commands are rigid; they also lack ex-

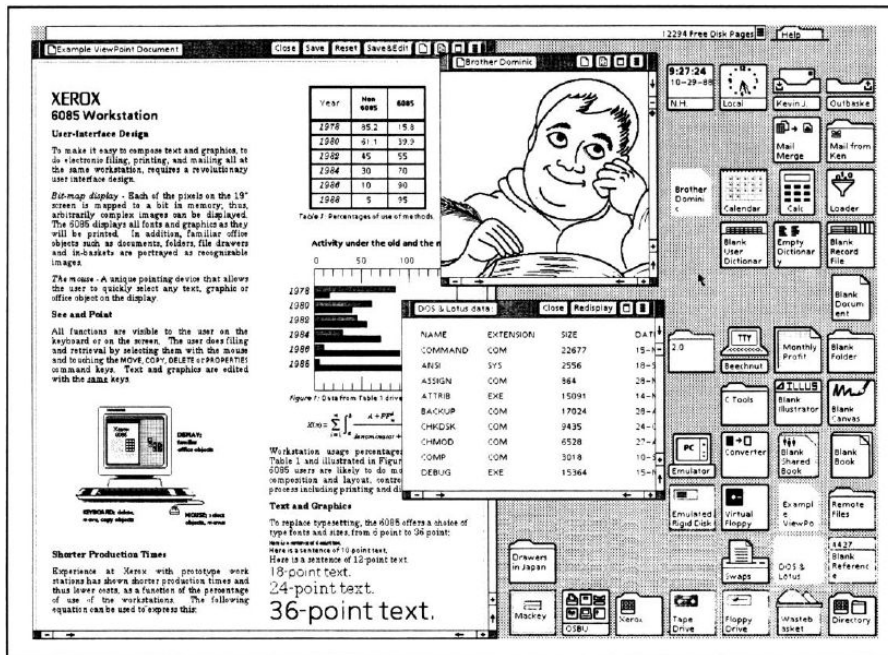


Figure 10: The Xerox Star User Interface [81, 82].

tensibility as users cannot create their own features or tools in their environment, nor share them with or import them from other users.

In the physical world, our tools use and workspaces organization is more fluid. We adapt our work environment to our current task, and sometimes use physical objects as tools in unexpected ways. We perceive different properties from objects, from their material, shape or structure, to infer how to interact with them. James Gibson [28] describes these possibilities as *affordances*. Affordances are the possibilities perceived by a subject, offered by the properties of an object, to act on its environment. This natural means of understanding one's environment possibilities is learned through the phenomenon of *perceptual learning* described by Eleanor Gibson [27]. Perceiving these affordances allows us to use physical tools to interact with our environment, and these tools become an extension of our body. We often use physical tools beyond their primary function [24], a phenomenon described by Renom et al. [70] as re-purposing tools, for example when someone uses a knife as a screwdriver because its material is hard, the shape fits a screw, and that is all they could find in their environment that fits these properties. This phenomenon is observed and described by Osuriak et al. [63–65] as *technical reasoning*, the capabilities humans have to focus on abstract laws based on the properties of objects rather than on their primary purpose, and how these properties would act on their environment.

Based on these premises, and to think beyond WIMP interfaces and applications-based systems, Beaudouin-Lafon [10, 11] introduces the

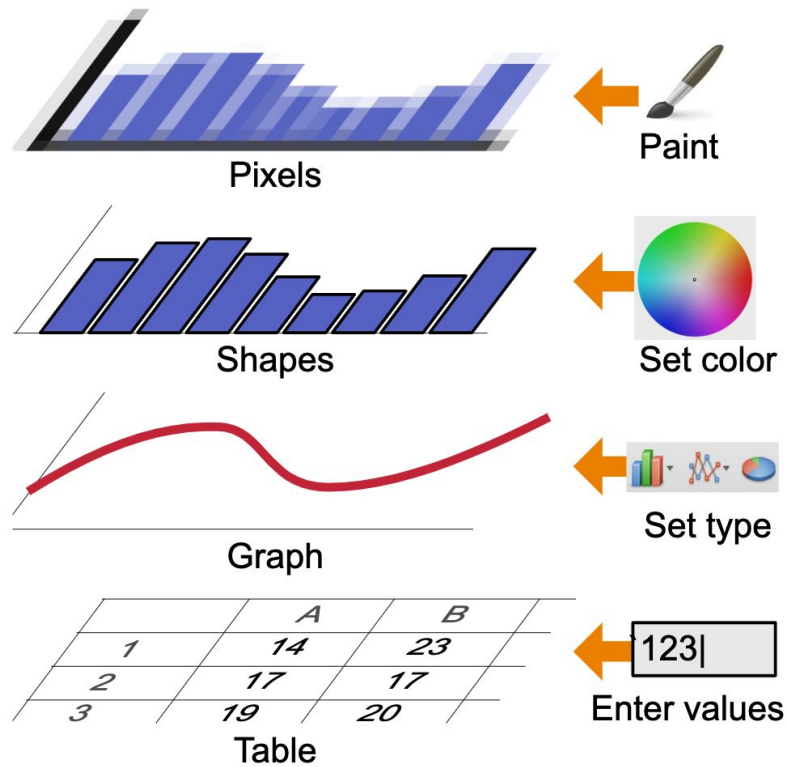


Figure 11: A rich substrate composed of different substrates, each with their own affordances, and associated instruments to operate on these substrates [10, 11].

model of interaction substrates and instruments. This model is based on the concepts of interaction substrates, which structure digital information, and instruments, which interact with these substrates. The goal of this model is to design interfaces that suit the capabilities of humans to understand and appropriate their environment. It aims to help design more flexible and extensible systems than the current WIMP paradigm.

An *interaction substrate* represents a digital object's affordances, it defines the constraints and properties of digital information. A single digital object can be composed of different substrates, representing different layers of information, each with their own affordances and with persistent relationships between one another. These collections of substrates represent a rich digital material to be interacted with (Figure 11). To operate on interaction substrates, Beaudouin-Lafon et al. [9, 13] introduce the principles of instrumental interaction. These principles describe an *instrument* as a first class object reifying one or several commands. These commands are polymorphic, i.e. they can operate on different types of objects, and their input and output can be accessed and reused later. Ultimately, instruments are interaction substrates that have the capabilities of acting on other

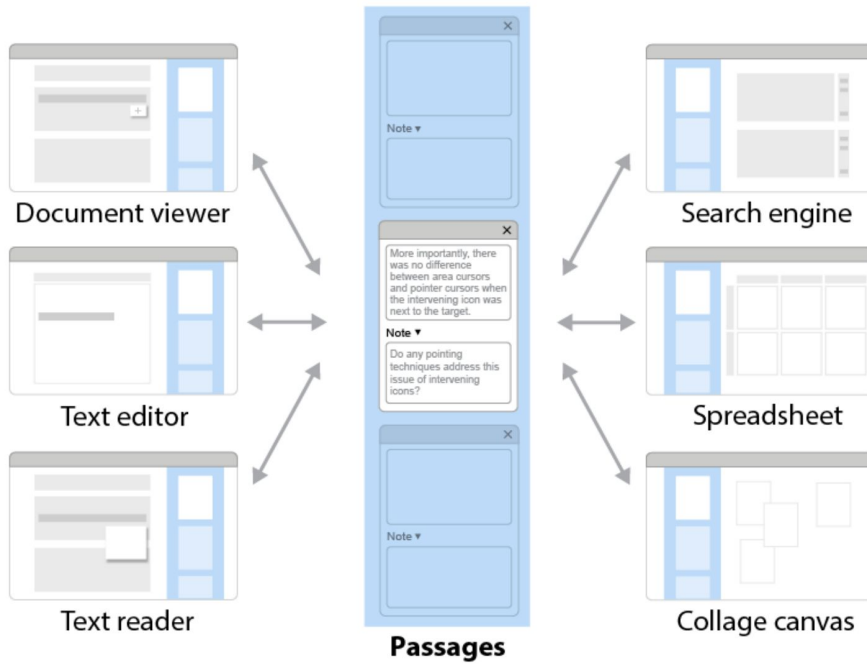


Figure 12: Passages [33] are reified text selections that can be manipulated, reused and shared across applications.

substrates. A digital environment composed of interaction substrates and instruments then provides more flexible interfaces to users, allowing them to reflect on which tools can interact with which object, depending on their goals, their preferences and the affordances they perceive.

This approach has already been explored to design new interactive systems. With Paper Substrates, Garcia et al. [26] design interactive paper components of different sizes that contains or control different typed data representations, here for musical compositions. These paper substrates allow users to link multiple representation layers of their data as modular and reusable paper artifacts to explore musical composition. Mixing interaction substrates and game design, Gustafsson et al. [30] introduce Narrative Substrates, using the principle of reification from instrumental interaction [9, 13] to represent players' traces as persistent and interactive gameplay elements and create unique content. In the context of text documents, Han et al. [32] designed Textlets, which reify text selection into persisting objects. Each Textlet has an attached *behavior* such as counting words, managing alternatives or matching occurrences of a search string. The goal is to help manage highly constrained text documents such as contracts and patents. With Passages [33], Han et al. also reify text selection as a first class object that can be manipulated, reused and shared across different applications (Figure 12), each time attaching a behavior depending on the applications' features. Passages allows to analyze and

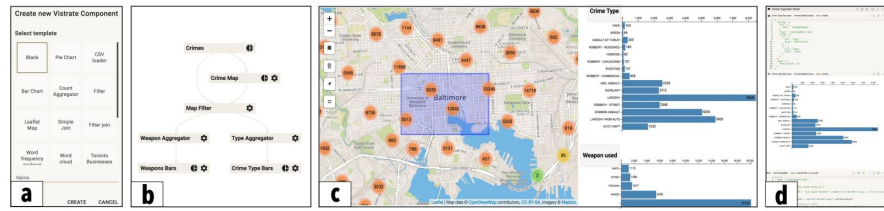


Fig. 3: (From left to right) **Component instantiation**: Existing component prototypes are available to be edited, thus, promoting reusability and extensibility. **Pipeline view**: This view supports configuration through interaction—drop-down selection—to create visualizations and interactively explore data without programming. In this example, a crime dataset from Baltimore, MD is visualized through a map and bar charts for crime type and weapons by aggregation. A filter component is added to filter the bar charts based on the selection on the map. **Dashboard view**: This vistrates view creates a grid layout for visual exploration of the data and annotation using rich text. **Development view**: The lowest level of abstraction for a vistrates in which a programmer can edit the code and create new visual analytic components.

Figure 13: The Vistrates' architecture [6].

compile text selections from different documents and supports knowledge workers in transitioning among applications and activities using these text selection substrates.

With their work on Webstrates [43], Klokrose et al. build upon Kay's [40] notion of personal dynamic media and introduce the Webstrates system, a prototype of shareable dynamic media. A webstrate is a web document object model (DOM) substrate. The system uses web technologies to provide an environment that satisfies key concepts that echo the concepts of flexibility and extensibility that we saw above with substrates and instruments. Shareable dynamic media is defined by Klokrose et al. as being "*malleable*": users can appropriate their tools and documents in personal and idiosyncratic ways; "*shareable*": users can collaborate seamlessly on multiple types of data within a document, using their own personalized views and tools; and "*distributable*": tools and documents can move easily across different devices and platforms. A webstrate is a shared collaborative object, and the changes brought to the DOM of a web page are propagated to the other clients of this page in real time. It also shares embedded code, allowing users to collaboratively tweak the application software behaviour. A webstrate can transclude other webstrates, allowing to customize users environment by importing new tools freely into it, blurring the lines between documents and applications. The goal of the Webstrates system is to be the basis for proposing more malleable software, and systems such as Codestrates [18] or Varv [17] demonstrate how to make it easier to develop for this kind of software. Other works have shown how Webstrates and this idea of malleable software are relevant in various use case. Among them, Videostrates [44] is a toolkit for creating real-time collaborative video editing tools. It explores fully programmable interfaces for video editing with examples such as real-time programming of blue screen special effects, and collaborative and synchronous video editing on different devices and tools. With Vistrates (Figure 13), Badam et al. [6] propose a component-based approach to data visualization. Vistrates allows to develop shared and distributed components that can be col-

laboratively modified and interacted with. The Vistrates environment includes a collaborative data pipeline in which Vistrates components can be simply plugged in to generate the desired visualization.

These theoretical works and their applications show the benefits of thinking beyond the boxes imposed by the domination of applications in our current interactive systems. In the physical world, we adapt our environment to our needs, depending on what we perceive is possible with the objects surrounding us, but this is hardly supported by our digital environments. With the interaction model of interaction substrates and instruments, and with the concept of malleable software, we can see that supporting such flexibility in interfaces is possible, as demonstrated by various use cases. For example, Passages [33] shows how to treat text as a substrate shared between applications, and Videostrates [44] shows how to interact on the same substrate with different instruments.

Interaction history tools suffer from locking history data in their own data silos, and users are not able to manipulate it with other tools, share it across applications and compile it together to make sense of it more easily. This is why I think that these concepts are important : to design better history tools that support users needs, we need to free history data from their applications and provide a flexible environment that lets users manipulate it with different instruments according to their goals. My research direction is therefore as follows: I first need to define a set of constraints and properties that define history data in general, i.e. an interaction history substrate, and then identify the specific needs users have to interact with it. Based on this, I can then design, implement and evaluate new history instruments.

In the previous chapter, the literature I reviewed showed that interaction history is not only about logging and analyzing one's past actions, but also giving material to the user to navigate through the timeline of digital contents. I showed that previous studies underlined both the importance of interaction histories in navigating through past actions on digital content and the lack of support to better understand the context of these past actions. Indeed, previous studies showed the importance of enriching timelines and history tools in general with meaningful contextual cues and providing the user with better means to understand the past.

In this chapter, my goal is to ground this understanding of history data manipulation in collaborative settings. Supported by the literature, I show why collaboration is a good case to study interaction history tools and the importance of enriching histories when sharing past actions with collaborators is needed. The goal is to describe how users are actively using traces in a specified collaborative use case. I want to understand to what extent their current tools allow them to collaborate around traces when they need to, and if the lack of flexibility I described in the previous chapter hinders their collaborative practices.

Considering related works from the literature and my own experience, I chose to study the case of collaborative writing. After describing the collaborative writing traces ecology, I detail how I conducted interviews and shared questionnaires with knowledge workers in order to gain insights to complement the current state of the art. I analyzed the gathered data using a mix of qualitative and quantitative analysis. I conclude and discuss the results that lead to the need for cross-application histories.

3.1 SETTING THE SCENE: WHY COLLABORATIVE WRITING?

In order to ground my understanding of user's behaviour and issues regarding history data usage, I focused my work on collaborative document editing. More precisely, I want to understand how people work with traces in order to coordinate their work during collaborative writing activities.

3.1.1 *Coordination*

Coordination is defined by Malone and Crowstone [56] as "*managing dependencies between activities*". As explained by the authors, this definition is inclusive in the way it could be used to describe coordination between any kind of systems, whether it be computational, biological, or human. In our case, focused on individuals, coordination can be understood as the means and strategies put in place to manage cooperation, collaboration and competition between collaborators for joint activities to run smoothly. What it means is that in collaborative settings, the ability to synchronize every actor on each of the various timelines generated by the tools they use is vital for collaborating efficiently.

This mechanism is demonstrated by Avdic et al. [5] with their two studies on collaborative writing and smart homes. They underline the importance of sharing knowledge of past actions for coordination. They explain that being able to trace past actions between users and digital or physical artifacts enables individuals to develop joint practices around these commonly owned and used artifacts by understanding how others interact with them. In other words, improving users' access to interaction history in collaborative settings tends to improve coordination.

This effect of aligning knowledge of past actions between collaborators on coordination is also shown by Hawkins et al. [34]. In a study on improving visual communication, they show that sharing traces of past visual communication between individuals makes them develop this shared knowledge and communicate better. In FileBiography [49], Lindley et al. describe a way to share knowledge about the propagation history of some content by enriching the shared content with different history elements such as the sharing, modification or annotation history. More than just a token to tell stories about content propagation, Lindley et al. show that a file biography also acts as a shared artifact between individuals, encouraging them to reflect on it as a social group. For the authors, history here acts as a tool for reflecting, learning, and collaborating among individuals from a given social group.

3.1.2 *Territoriality*

Coordinating is also a matter of understanding the role of each other, and respecting their spaces in order to better organize or avoid conflict. Individuals need to understand their role in a collaborative project, so that they can feel that they have some control over a part of the digital content, and in some cases some privacy too [47]. This echoes the concept of territorial functioning described by Taylor [83]: *“Territorial functioning encompasses a class of environment-behavior transactions concerned with issues of personal and group identity, cohesiveness, control, access, and ecological management”*. According to Taylor, territorial functioning is a group-based human behavior that aims at enhancing relationships by distributing among individuals the space that they can have control over, in order to better collaborate. This sense of territory, also called *“Territoriality”* by Bakker and Bakker-Rabdau [7], is not only limited to physical space, but also includes personal thoughts and attachment to tasks or roles. As such, they define an individual territory as *“those areas in which he has special expertise, shows initiative, takes responsibility, where he has control”* [7].

Several works underline the importance of understanding how territorial functioning influences collaboration in collaborative writing settings, especially Larsen et al. [46, 47] who discuss how it can help design better tools to support coordination. Olson et al. [62] and Wang et al. [85, 86] also studied this effect in collaborative writing settings. Not only did they show that even with synchronous tools people do not really collaborate in the same space of the shared digital content given their roles in the team, but also that using traces they can better organize and coordinate as a group. With Docuviz [85], Wang et al. enrich the document’s modification history by providing a visual representation of the changes within specific periods of time that inform users about collaborative patterns on a shared document. This helps them organize their work around history data by reviewing their past activities over time and re-discussing roles and territories and thus their collaborative process.

3.1.3 *Ecologies*

Territoriality is a well-known phenomenon and a de facto behaviour allowing individuals to better coordinate on common digital objects such as text documents. But other works such as Bergman’s study on personal information management [15] showed that individuals tend to distribute their activity, supposedly occurring on one object, across different applications, storing and accessing data in different formats, from different places in their digital environment. That fact is later underlined by Ledet et al. in their work on territoriality [45, 47], describing the collaborative writing process as fragmented, in

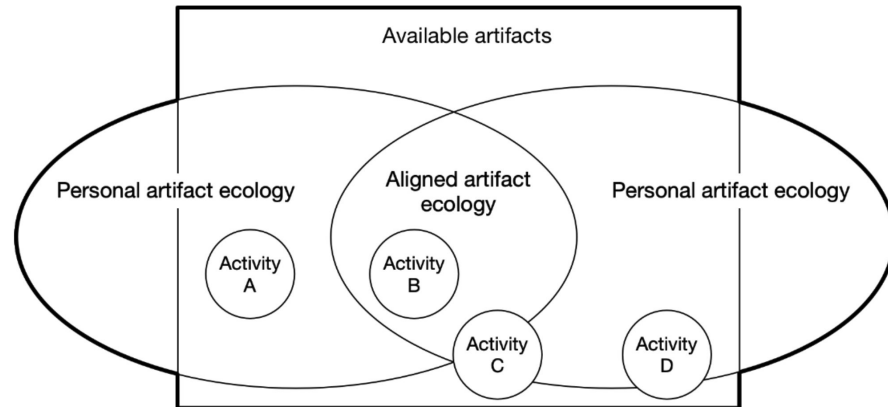


Figure 14: Overview of artifact ecologies in collaborative writing. Figure from [48].

terms of spaces in the text document, but also among the different tools and devices of each individual.

This observation echoes the concept of ecologies, or artifact ecologies. Gibson [28] uses the term “*ecology*” to describe how our physical environment impacts our individual visual perception. Based on Gibson’s work, Jung et al. [38] define a personal artifact ecology as “*a set of all physical artifacts with some level of interactivity enabled by digital technology that a person owns, has access to, and uses*”. This definition clarifies the fragmentation of human activity in digital environments through the multiplicity of accessible devices, and is appropriated by works such as Bødker and Klokmoose’s [16] investigation of how individuals use technology ensembles. Rossitto et al. [72] extend this investigation on collaborative uses and do not limit the personal artifact ecology to only physical devices, but also to software applications. The authors draw the attention to the fact that the interconnection between the different artifact ecologies from the different actors collaborating on a common object have an impact on how they orchestrate collaborative work and how they coordinate their use of their different tools and devices.

By doing so they introduce the concept of “*aligned artifact ecologies*”, later reused by Larsen et al. [48] when investigating collaborative writing artifact ecologies (Figure 14). They state that “*in collaborative writing, the potential artifact ecology is the sum of co-writers’ personal artifact ecologies, from which they negotiate an aligned artifact ecology consisting of the tools and applications to be used in the work*”. Taylor [83] and Bakker and Bakker-Rabdau [7] defined territorial functioning, and how people manage territories on common objects [47], and Larsen et al. [48] definition of an aligned artifact ecology shows how people manage

territories in environment composed of multiple artifacts. But collaborative work is not limited to artifacts and tools accessible by everyone. For example, Nouwens et al. [61] identify that this concept of ecologies also impacts the use of communications apps. Depending on which individuals, and for what purpose they are communicating with each other, users will prefer using a specific communication system, or channel.

To conclude, even when centered on a common object, traces of collaborative activities will be left in different places in the available artifact ecologies of the collaborators, making it hard to keep track of their common timeline [5]. Coordination in that setting needs to be supported by cross-application history tools and instruments and by sharing interaction history data across applications [69], which is rarely supported due to the lack of flexibility for interaction history designs.

3.1.4 *The Collaborative Writing Traces Ecology*

When collaborating around a common object, users establish coordination strategies such as territoriality to organize their work. This behaviour is widely observed in collaborative writing activities [47, 86]. Based on Jung's et al. [38] observation of how artifact ecologies play a huge role in collaborative activities [48, 72] and how accessing traces left in these ecologies may be vital for collaborative work to run smoothly [5, 85], I chose to investigate how people coordinate in collaborative writing activities using traces of their activity.

Collaborative writing environments offer great diversity in terms of the historical data they generate. Users have to deal with both their personal artifact ecology and an aligned artifact ecology based on their collaborative process as a team. To describe this phenomenon, I introduce a framework for collaborative writing applications and traces ecologies (Figure 15) based on the work of Larsen et al. [48] and their description of the collaborative writing artifact ecologies.

With this framework, I make several statements. First, the aligned application ecology does not only involve an aligned collaborative editing tool with integrated facilities, such as *Google Docs*¹ or *Overleaf*², but also communication tools such as email and *Slack*³, shared storage tools such as *Dropbox*⁴, and versioning systems such as *git*⁵. It is complemented by the personal set of tools used by each participant

¹ <https://docs.google.com>

² <https://www.overleaf.com>

³ <https://slack.com>

⁴ <https://dropbox.com>

⁵ <https://git-scm.com>, <https://github.com>

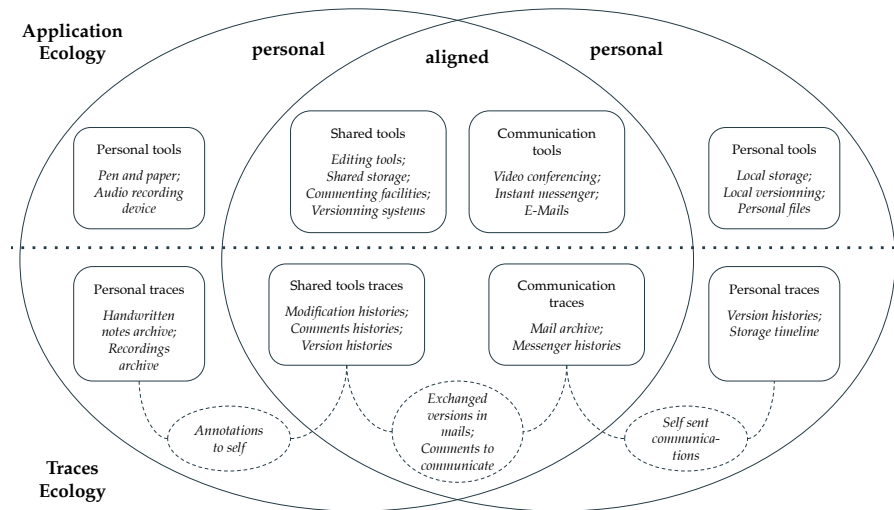


Figure 15: Collaborative writing applications (top half) and traces (bottom half) ecologies. In the top part, the rectangles represent the main categories of tools used during collaborative writing activities. In the bottom part, the rectangles represent the main categories of places where traces about a collaborative document appear and contain typical physical or digital objects holding the traces. The ovals represent traces that belong to the two adjacent categories.

based on their preferences and habits, such as personal storage tools and personal editing tools. Second, traces are left in each of these tools and applications, whether they are integrated into the editing environment, such as modification history and annotations, or not, such as shared documents related to the activity and versions in a version control system. The traces ecology also includes personal traces left by each participant's personal set of tools, and traces of communications among team members. Finally, the different categories of places may overlap. For example, annotations in a document may be part of the traces of a discussion, and communications could be used as part of one's personal space, for example sending notes to oneself by email. This means that the place where a trace is left does not always correspond to the meaning it conveys.

This framework serves as a focus for the study described in the next section. It is used to better understand user breakdowns when interacting with history tools in collaborative writing environments.

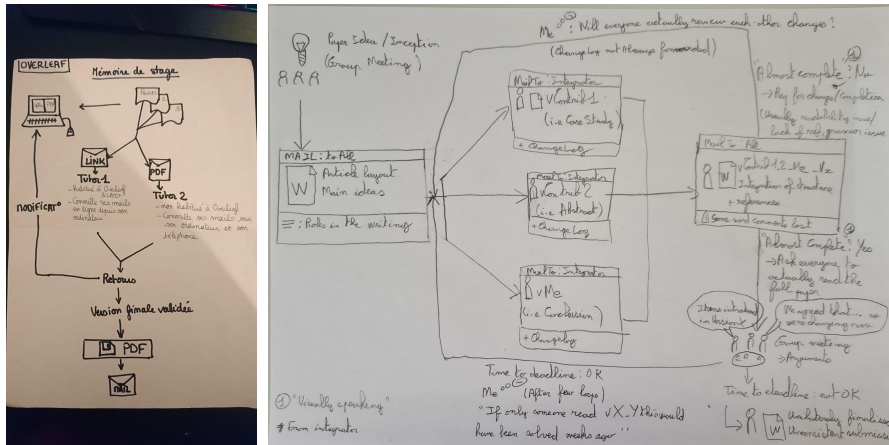


Figure 16: Two examples of drawings created by interviewees, from P2 (left) and P3 (right).

3.2 STUDY 1 : INTERVIEWS WITH KNOWLEDGE WORKERS

Based on the framework I depicted in figure 15, and the literature about traces usage in collaboration, the study I describe next addresses the following two research questions:

- RQ1 How do people use histories to coordinate and keep track of the timeline in a collaborative project?
- RQ2 What are users' actual issues regarding history usage in a collaborative setting?

3.2.1 Method

To answer RQ1 and RQ2 in this first study, I used a mixed approach. I conducted story interviews [52] with 12 participants that I analyzed using thematic analysis [19, 21].

PARTICIPANTS. I recruited 12 participants for the interviews: 8 men, 4 women; 6 aged 18-30, 2 aged 30-40, and 4 aged 40-50. 6 participants were from academia (one HCI Ph.D. student, one biology Ph.D. student, one medical physics Ph.D. student, two biology research engineers, and one biology associate professor) and 6 from industry (two IT engineers/technical managers, two IT project managers, one scientific writer, and one lead UX designer).

PROCEDURE. I asked participants to tell me the story of the last document they edited collaboratively. I made them go through their whole process, focusing on history data usage when necessary. As the interviews took place during the COVID-19 pandemic, they were con-

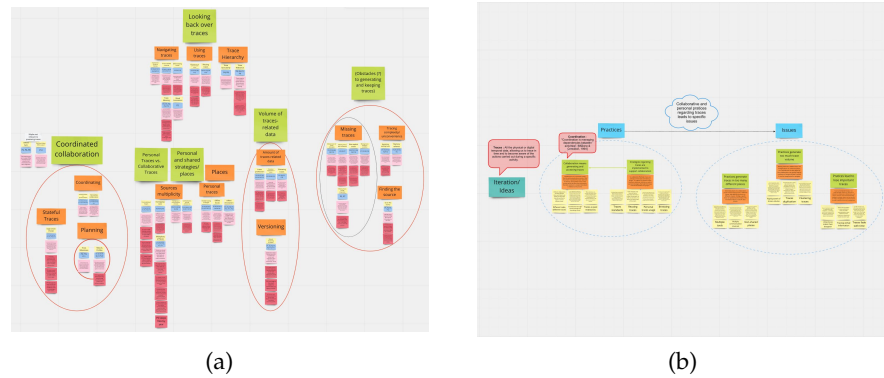


Figure 17: Two iterations on the thematic analysis using a Miro board. (a) is an early stage where we started coding and grouping codes and quotes; (b) is a later stage where we started grouping and defining themes.

ducted remotely, through video conferencing. The interviews were audio- and screen-recorded and transcribed for analysis. The set-up being unconventional and to avoid the discomfort related to the situation, I asked them to bring a quick and dirty drawing representing their writing process (figure 16). It helped me have a physical artifact to refer to when the discussion was losing fluidity, but also for them to reflect on their past actions.

ANALYSIS METHOD. I conducted a thematic analysis [19, 21] to analyze the interviews with the help of Miguel Renom, a fellow researcher. We worked together to identify recurring themes, first iterating on top of each other’s categorization, and then working together to merge close themes and discuss disagreements. As this process was remote too, we used the web application *Miro*⁶ to help us brainstorm and organize codes and themes (figure 17).

3.2.2 Results : Thematic Analysis

The thematic analysis led to five main themes, divided into two categories according to the two research questions. Table 1 is an overview of these themes, further described in this section.

The first category is about participants’ collaborative practices towards traces and their strategies for using them to coordinate with other team members (RQ1). Interviews showed that users re-document traces to regroup them and facilitate coordination (Theme 1/T1), and that they set up conventions to better differentiate and understand traces (T2). These two themes express two different ways of ma-

⁶ <https://miro.com/>

Category	Theme	Key Elements
Practices (RQ1)	T1: Users re-document traces to group them and facilitate coordination	Users group traces to avoid bloated histories Users group traces to share progress status Users gather information from different histories
	T2: Users put personal or collaborative norms in place to better differentiate and understand traces	Making important traces stand out Emphasizing the meaning or context of past actions Customizing traces to browse them more easily
Issues (RQ2)	T3: History data are left in too many places around the object of the collaboration	Traces are left in both personal and aligned ecologies Different roles, different tools, and no shared history Various communication tools and no joint code of conduct
	T4: Too much historical data is generated during collaborative activities	Large number of operations, large amount of traces Traces duplication due to a complicated access No control over the histories' granularity
	T5: Important historical data get lost over time	Traces lost due to tools erasing them Traces' meaning is lost over time due to users' memory

Table 1: Overview of the five identified themes.

nipulating traces and are each a way to better understand interaction history and better coordinate on a shared document. The strategies put in place to differentiate and better identify important or related traces can also be used to regroup traces according to these differentiating attributes.

The second category is about the main issues that participants face when accessing history data, sharing it with others, or simply using it, and which prevent them from progressing effectively as a team on their collaborative project (RQ2). In this second category, I identified three themes. First, history data are left in too many places around the object of the collaboration (T₃), and traces are left in a scattered traces ecology amongst individuals collaborating together, making it hard to keep track of the progress and synchronize timelines. Then, I also identified that even within a single history, too much history data is generated during collaborative activities (T₄), especially during intensive periods of collaboration. This amount of generated data is often not controlled by the user and justifies the practices I identified in the first category aiming at summarizing histories as a new simplified artifact. The last issue I identified is that important history data get lost over time (T₅), whether because of the lack of persistence of data itself, or because their meaning or the context of the trace creation did not persist.

3.2.2.1 *Theme 1 : Users re-document traces to group them and facilitate coordination*

The meaning of the word re-documentation here is borrowed from Yahiaoui et al., and means to “appropriate the content, shape or scope of existing documents according to a personal vision” (translated into English from [90]). In our case, it applies to traces rather than only

documents, and their reshaping into a new artifact that may be personal but also shared among individuals.

Traces re-documentation can take different forms. It can be a grouping of traces in order to summarize the progress made between two different milestones of a project and make it understandable. For example, P2 says she compiles every annotation made on the shared document, sent by e-mail or simply said out loud during a face-to-face discussion, and writes an email to her collaborators before a meeting so that everyone is on the same page. With the same idea, P3 collects important modifications made to the shared document before a milestone so that it summarizes important progress and avoids the “bloat” due to minor modifications not worth discussing. He then uses this compilation to set-up an agenda for the next meeting.

These groupings can also take the form of specific storage places for important and related traces. For example, P10 explains that her team has a specific shared folder for storing the version of the shared document each time they cross a new milestone. As a more personal strategy, P12 says that she sends herself emails every time she wants to add an item in her “to-do” list. By doing so, she can regroup this information in a single application.

3.2.2.2 *Theme 2 : Users put personal or collaborative norms in place to better differentiate and understand traces*

Interaction history data logged by the same source application does not always have the same importance or the same meaning. To better understand traces left by other individuals in that case, standards must be set in order to easily identify what’s behind a past action.

The nature of a trace could easily be identified through the role of the individual who left it. Team members often have different roles, leading to different traces left on the project, and different interests in accessing history. For example, P2 said that she is the only one modifying the shared document, while the other members only leave or send her comments and suggestions. What it means for her is that when she receives a new version of the document, she knows it is not a modified version but only an annotated one.

But sometimes, there is a need for more elements to understand the meaning of a trace. For example, two changes in the document may not actually both be modifications of its content. One could be a question directly typed in the document from one team member to another and logged by the application as a modification. For example, P3 explains that many of his team members do not use the annotation tool integrated in their text editor. Instead, they add their comments directly in the text and change the comments’ color based on the

importance of the annotation. But even using dedicated tools does not prevent users from needing such additional information. P9 explains that to distinguish an urgent modification from a minor one, his team adds labels at the beginning of the annotation. To specify the content of a trace, an overused but often necessary strategy is also to have naming conventions for shared documents, versions, or even emails. It makes it simpler to find a specific trace in the mass of information when, as P12 explains, it is easier to find a trace with keywords rather than with the time period with which it is associated.

3.2.2.3 *Theme 3 : History data are left in too many places around the object of the collaboration*

The issues identified within this theme are strongly related to the observations we made in the first theme (T1) about how people like to re-document traces, and the fact that traces in a collaborative writing setting are scattered inside the collaborative writing artifact ecology I described earlier. Regarding the example of P2, she has to collect annotations from different places such as emails, LaTeX files and annotated PDFs to compile them and it takes her time, but also mental effort to make sure she does not miss any trace.

Within the same team, people can also use different tools to edit the same document, to annotate it or to share files. The multiplication of histories containing traces can make it difficult to synchronize and integrate timelines. For re-documentation purposes and even just for browsing information, it makes it hard to find all the relevant information that users want to compile. For example, P9 uses one tool to edit the collaborative document (Microsoft SharePoint), and another one to annotate it (Ideagen PleaseReview). The reason is that the members of her team do not have the same role and responsibilities, and in that case, not the same access to specific tools. This setting makes some traces invisible depending on the tool used and the users have to know that a piece of history exists to find it.

The number of different communication channels also plays an important role in the difficulties users face when coordinating. Communication among team members uses various channels such as email, instant messaging or even in-document conversations, making it harder to find what has been said about a specific issue. For example, P12 explains that she may “*know [that] a file has been shared*” but there are numerous communication channels and no “*joint code of conduct*” about what to share and where. It results in a loss of time spent looking for that piece of information in her past communications.

3.2.2.4 *Theme 4 : Too much history data is generated during collaborative activities*

Large numbers of operations on a collaborative object or the number of communications generate large amounts of history data, and it quickly becomes cumbersome to navigate through it. People also tend to duplicate some traces such as document versions or e-mails to consolidate them or to have a different way to access them, adding to the volume of the traces and creating redundancies. For example, P6 sends himself versions via email in addition to his shared versioning system and his local files, to access it more easily in case he loses access to his computer or his shared storage. Another example is P1 who is afraid of losing important traces and does not trust her shared tools to save them, and thus makes periodic local backups of the shared document.

Overall, current tools do not let users control the granularity of historical data and do not support clustering or summarizing over a given time period. These operations are often performed manually by the users, who spend a large amount of time doing so, as highlighted in the first theme (T1).

3.2.2.5 *Theme 5 : Important history data get lost over time*

History data can simply start disappearing or lose meaning, like traces that fade with time. This often leads to confusion, misunderstandings and to a heavy reliance on the user's memory. Some tools do not ensure persistence for data such as annotations. For example, P3 lost annotations after merging two different versions of the same document, and he had to go through the previous versions of the document to find and re-annotate the document himself with missing information. But at some point he missed an annotation asking for a major change in the document, and the associated task was not completed for weeks because of that. Some tools can also be misleading in the way they integrate annotations. P7 says he lost some comments on the document after indicating that the issue had been addressed. He could not access the history of annotations even if he wanted to because it was not built into his tool.

However, losing the meaning of data over time is mainly due to the lack of tools to help users recall the context when the data was created or the motivation behind the user's actions. For example, P7 often cannot remember why he left annotations which meaning is not clear anymore.

3.3 ADDITIONAL INSIGHTS WITH QUESTIONNAIRES

From the interviews, we've seen that participants improve collaboration by adopting coordination strategies requiring users to share relevant historical data. This raises the question of what tools and historical data people are willing to share in the context of collaborative text editing.

Interview participants also reported browsing historical data from different sources to make sense of a project's timeline. However, these sources are not always compatible with each other. Sharing historical data with other team members may also mean sharing data that the other team members do not currently have access to. This raises questions about the relevance of cross-application history systems and tools.

Hence to complement my results, I chose to distribute a questionnaire to a new set of participants.

3.3.1 *Method*

PARTICIPANTS. The questionnaire was distributed to 27 participants through research laboratories and HCI community mailing lists. I collected $n=27$ responses from knowledge workers, mainly from the French HCI community: 14 men and 13 women; 18 aged 18-30, 4 aged 30-40, 2 aged 40-50, and 3 aged 50+. The questionnaire was administered online through the Framaforms open-source surveying tool⁷.

PROCEDURE. The questionnaire was composed of five-point Likert items with questions where respondents evaluated their privacy concerns towards sharing their historical data with relevant members of their team and their need for a cross-application history system.

The first part of this questionnaire asked users about their collaborative writing habits. It makes sure they fit the profile of knowledge workers often involved in collaborative activities. This section included questions such as:

- What was the most recent collaborative writing project you participated in?
- What tools did you use to perform this collaborative work?
- What means of communication did you use to exchange with the different stakeholders of the project?

In the second part of the questionnaire, respondents evaluated their privacy concerns towards sharing their history data with relevant

⁷ <https://framaforms.org/>

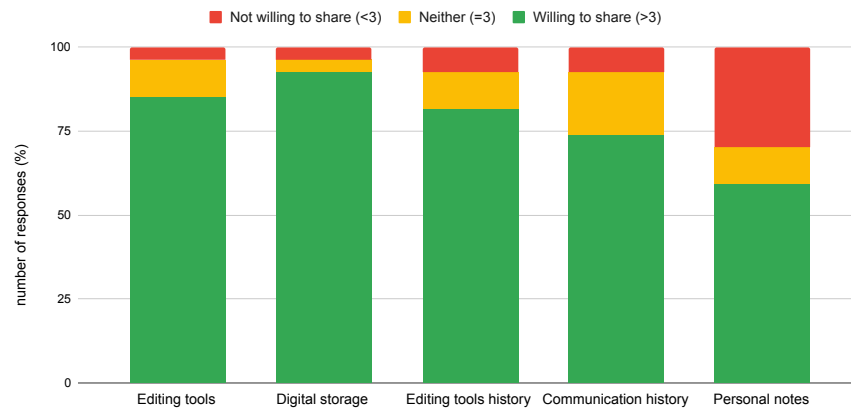


Figure 18: Willingness to share tools and histories (n=27).

members of their team, and their need for a cross-application history system⁸. This section included assessments such as:

- It would have been necessary for my current tools:
 - To be able to synthesize the progress of my team on a project at key moments.
 - To be able to visualize all activity traces in a single space to be able to navigate through them.
- During a collaborative writing project:
 - I am willing to share in real-time data such as my modification history or my comments on a shared document.
 - I am willing to share personal material such as my written notes, or my summaries of meetings about the project.

ANALYSIS METHOD. Questionnaire responses were collected through the web-based form into a data table. I analyzed the questionnaire data with summary statistics for each question.

3.3.2 Results

3.3.2.1 Willingness to share tools and histories.

Regarding the assessments about privacy concerns, I asked respondents about their willingness to share specific tools or data, including shared editing tools and their history, shared digital storage, and shared communication history. I also asked about their willingness to share their personal traces about the project, such as personal notes. Figure 18 shows the scores for each of these items.

⁸ The questionnaires are available in Appendix A.1.

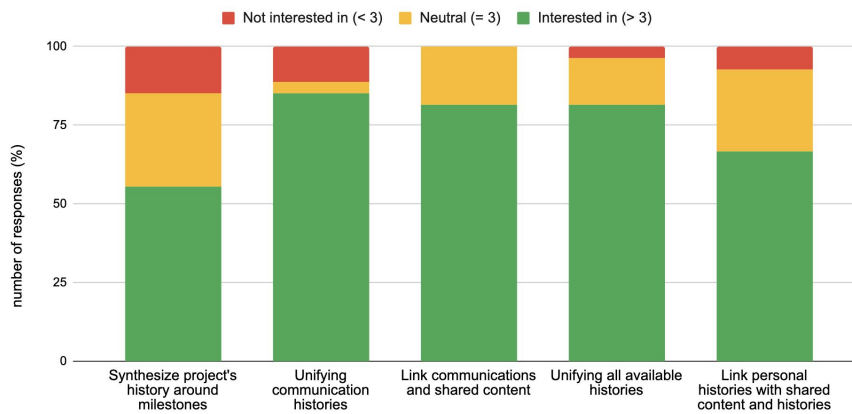


Figure 19: Interest in cross-application history tools (n=27).

Results show that most respondents are willing to use shared and synchronized editing tools (85%; 23/27) and to share a common space to store all their common work (92%; 25/27). To complement their shared editing tools, they are also willing to use the associated facilities to share historical data (81%; 22/27). 74% of the respondents (20/27) are also willing to share the history of their communications about a common project, and more than half (59%; 16/27) are willing to share their personal notes.

While high, these scores are not unanimous, therefore users should remain in control over what data is shared or not, especially potentially sensitive information such as communications and personal notes. Sharing should require user consent so that users can choose what information to share, and who to share it with.

3.3.2.2 Interest in cross-application history tools.

As stated before, considering the need for users to regroup histories from different places and thus create new history artifacts to be better understood by their whole team, I wanted to probe users' interests in cross-application history tools before drawing further conclusions. Figure 19 shows each item's scores.

A majority of respondents (55%; 15/27) are interested in the possibility of synthesizing a global project's history around specific milestones. Most respondents expressed their interest in unifying their communication histories (85%; 23/27) and in linking their communication histories with shared content (81%; 22/27) for cross-referencing. Furthermore, 81% (22/27) expressed their interest in unifying all histories they have access to during their project, and 66% (18/27) in linking their personal histories with shared content and shared histories for cross-referencing.

3.4 CONCLUSION AND DISCUSSION

In this chapter, I chose to investigate collaborative writing as a typical case of collaboration in order to shed light on the use of history as a critical component of coordination. I supported these claims with related work from the literature, and introduced the concept of collaborative writing traces ecologies. In order to better understand users' practices (RQ₁) and issues (RQ₂) towards trace usage in collaboration, I conducted story interviews which I analyzed through a thematic analysis, and complemented by a 5-point Likert-scale items questionnaire.

Regarding users' practices around history usage (RQ₁), I found that many strategies take place outside the scope of their history tools. Users want to re-document traces, to create new artifacts from their interaction histories (T₁), to make it more understandable for them and their team. In complex traces ecologies, traces of the collaboration are left in numerous histories. Re-documenting requires browsing across different histories, merging and extracting relevant information and compiling it as new content, in a new text document or email. The goal for users is to have comprehensive support to understand and communicate a specific state of the object of the collaboration and to organize, to coordinate around it. But current history tools are designed to monitor specific parts of a project's history, and the lack of interconnection with other tools' histories hinders the users' ability to use history data effectively. On top of the problems of regrouping traces together is the matter of differentiating traces from one another (T₂). Within the same histories users may need to indicate that a trace is more important than another one, or to emphasize the meaning behind a trace, or to add contextual cues to better understand it. To do so and since their tools do not allow them to add this information to the history, they develop techniques to customize traces such as color coding or name coding. Overall, these findings tell us that history data are not only defined by simple information items such as source, authorship or time, but also by the intention behind the user's actions and the motivations behind an event that occurred. In the end, and with the two main interactions with traces that I identified, it requires a lot of additional work from the users. It is not sustainable in the long term, and ultimately leads to coordination issues because current tools often do not support such appropriation of history data.

I identified three main issues about trace usage hindering coordination (RQ₂). I found that the number of applications and thus the histories that users have to deal with to understand their team's progress on a common project is detrimental to their work efficiency and coordination (T₃). Users have to navigate data in numerous places, which requires more work to identify relevant history data and risks leaving

one behind and reducing collaboration efficiency. To find data more easily, users even have to sometimes replicate history data to save it elsewhere and avoid being drowned in a deluge of data caused by intense collaboration. The impossibility for users to control the granularity of logged interactions also makes them shape traces as markers of the project's progress to refer to and understand relevant time periods of the collaboration. But in doing so, they may just add to the amount of data to deal with, and the number of places where data are stored (T4). Finally, history data are not just timestamped data, they also convey meaning. To understand a past action, one has to understand the context of its creation, the intention behind the action, and the relationship between it and the other actions performed in the past. This underlying information is often not consolidated and made persistent by history tools' features and therefore is likely to fade with time (T5).

Related works on shareable histories already showed the benefits of regrouping interaction histories to tell a story and better understand past actions [49, 69]. In my personal statement about how traces ecologies play a huge role on users ability to coordinate, I pointed out the difficulties of supporting this kind of shareable histories in the current collaborative environment where history data are scattered amongst rich and complex ecologies. The empirical findings from this first study tend to show that, while needed by users to support their collaborative practices, cross-application history manipulation is widely unsupported by current tools. It remains a demanding and cumbersome activity they need to perform. Based on these insights, I was able to gather knowledge workers' wishes and concerns in sharing relevant traces amongst team members. They all showed interest in such ways to improve interaction histories. Because this lack of flexibility from current interaction history tools hinders coordination and users' global timeline awareness and understanding, I want to introduce a new approach for designing interaction history systems, tools, and objects.

TOWARDS CROSS-APPLICATION INTERACTION HISTORY SYSTEMS

In the previous chapter, I reported on an empirical study that studied knowledge workers' habits toward interaction history usage in collaborative writing activities. It showed that looking at a single history does not help users understand the timeline of events, especially when multiple artifact ecologies are involved. Users need historical data to be consolidated across applications and shareable with other team members. The results of this first study, combined with insights from the literature, leads me to conclude that history data should be shared across applications. Users should also be able to manipulate them beyond the scope of said applications.

In order to fully support users' practices, historical data and collections should be independent from the sources that generated them. My solution is to reify [9, 13] historical data into first-class objects that users can readily manipulate with appropriate tools. By separating historical data from the applications that generated them, I want to enable any application to access the global history and to make it possible to design new, specialized history tools.

In this chapter, to support such flexibility in designing interaction history systems, I describe the first iteration of a unified history data structure which I called a "Step". Through Steps, I define the common attributes shared by interaction history data, to be later accessed by users and their instruments to manipulate their history. Then, I describe OneTrace, a proof-of-concept cross-application history system for managing Steps. I show how I implement a first version of this system, and how applications and instruments in this environment log and interact with Steps.

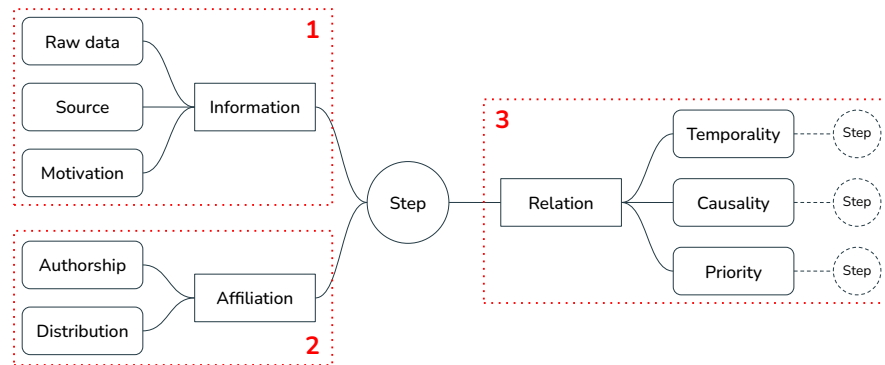


Figure 20: The three main components of a Step: Information (1), Affiliation (2) and Relation (3).

4.1 STEPS : A UNIFIED DATA STRUCTURE FOR INTERACTION HISTORIES

Previously, I argued that users need history data to be shared across applications and with other members of their team when relevant. To this end, we need a way to unify interaction history data by specifying common attributes that identify it. This will lead to the definition of a unified data structure that describes any type of trace and can be used by any available interaction history tool. First, this structure will lower the constraints for users to share history data amongst team members by not having to care about compatibility issues. It will also provide users with freedom in their choice of how they want to manipulate their history data, and with which instruments. Finally, it will give more freedom to designers to focus on how users interact with history data and provide them with meaningful tools, rather than taking care of the constraints that prevent data sharing among applications. Based on the results from the interview study, I want this new approach to: support data shareability across users, thus improving their coordination on common projects; support creating relationships among history data across applications, thus enhancing history recall and understanding; persist history data outside of their source application; and provide ways to capture the context of past actions.

In general, history is a succession of steps that have been taken to reach the present state of the observed context. I therefore call the common model encapsulating the structural data shared by all history data a *Step*. I describe a Step as having three components (Figure 20):

INFORMATION: It contains the *“what”*, that is the content of the action performed by the user, such as a text modification, the insertion of an image or the description of their action; the *“where”*, that is the application and/or document in which the user action was performed, the source that logged the action; and the *“why”*, that is the motivation behind the user’s action.

This component lets history data stand alone, with the added value of having information about its author’s intention. Note that while the *“what”* and *“where”* can be usually provided automatically by the source application or document, the *“why”* must typically be provided by the user. That fact implies designing tools specifically to allow such customization of history data, which is a need often highlighted in my thematic analysis.

AFFILIATION: It contains the *“who”* and the *“whom”*, that is the user who performed the action and which users the data should be distributed to so they can access and manipulate it.

As we have seen, the notion of roles is really important in collaboration, and to understand traces. Thus, users should be able to share their histories with the relevant collaborators and we need to support this notion of roles in histories.

RELATION: It contains three relationships that tie together the Steps:

- *Temporality* describes the temporal ordering of Steps based on the timestamp of each step (the *“when”*), resulting in a single timeline of events.
- *Causality* describes causal relationships between Steps, for example when one event occurred because of or in response to another event. This makes it possible to consolidate the context of past actions and to cluster history data into meaningful groups.
- *Priority* describes the level of importance of a Step, letting users focus, for example on higher-level tasks or on historical data at a higher level of granularity by using important steps as milestones.

While Temporality can be determined by the system, I assume for now that Causality and Priority have to be set by the user, with dedicated tools to be designed. Supporting such relationships would help design new interaction history tools by seeing beyond temporality and support users in their understanding of the timeline of events, or their re-documentation activities.

In summary, Steps are self-contained objects that can be used independently of their source application. Relationships can tie them together in a variety of ways, not limited to the traditional timeline

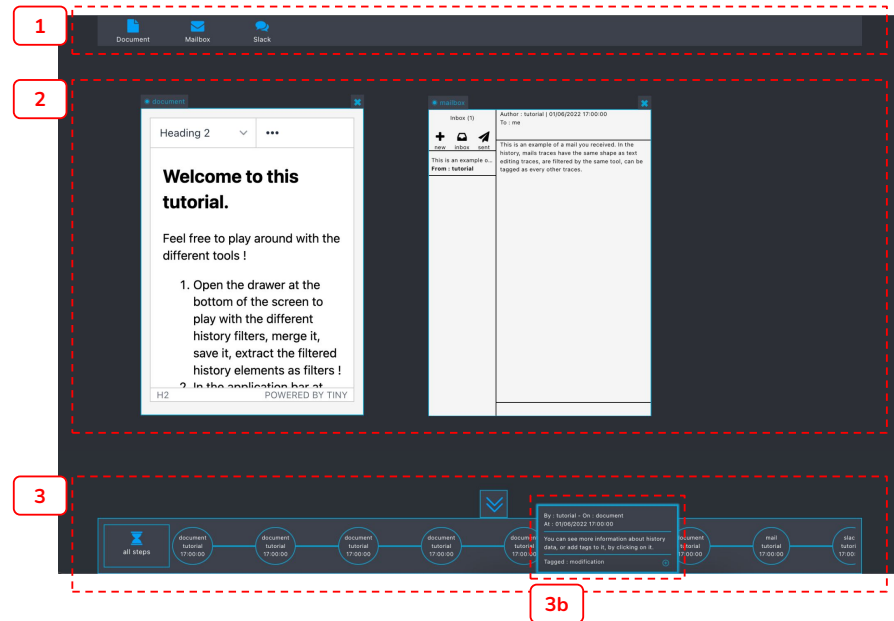


Figure 21: The *OneTrace desktop* features (1) an application dock, here with a collaborative text editor, a mailbox, and a messenger, (2) a workspace with two open applications and (3) the main timeline where users can (3b) magnify the Steps.

or list of commands that are common in current applications. Steps are also extensible: the aforementioned attributes that we identified thanks to our first observations could be complemented with more specialized attributes. Applications that manipulate Steps should simply leave untouched the attributes they do not recognize. Steps therefore provide a sound basis for creating cross-applications history systems.

4.2 THE ONETRACE SYSTEM : A CROSS-APPLICATION INTERACTION HISTORY LOGGING SYSTEM

To manage Steps, I first wanted to implement a basic system that would be the common foundation for the future interaction history tools I want to design. As a proof-of-concept, I developed OneTrace, a client-server system that is a shared cross-application interaction history logging system implementing my concept of Steps. To interact with this system, I developed the OneTrace Desktop web-application. This application simulates a collaborative desktop environment (Figure 21). I wanted it to be as generic as possible so I could integrate new tools and instruments to manipulate interaction history data in collaborative writing use cases. Any action performed by any user on any application on the OneTrace Desktop is logged by the OneTrace server by creating a new Step. The content and granularity

of the logged interaction can be specific to the application, but all Steps have the same structure as I described above. The OneTrace system also shows limitations in that it does not fully implement synchronous collaboration yet. But the setting allows for simulating such conditions for potential user studies.

Since this is a proof-of-concept prototype, I limited the implementation to the minimum set of features needed to simulate a collaborative setting. The user interface features three main parts. The first one is an application dock at the top of the screen. Double clicking on an application's icon will open it in the second part of the environment which I call the workspace. The third one is a global timeline of the actions performed by the user and their teammates, in the collaborative context. Each application in this environment automatically logs Steps when users interact with it. Steps are stored on a server and displayed on each user's global timeline in real-time. This main timeline is contained in a drawer which users can open or close at their convenience.

In the global timeline, the visible Steps are those authored by the user as well as those shared with them, i.e. those whose Affiliation (the "*Whom*") includes the user. Clicking on a Step in the timeline magnifies it and displays more information, such as its detailed content. A tagging system lets users manually add more information to a Step, a simple yet straightforward feature to add information in the Motivation sub-component of the Step (the "*Why*").

The front-end of the OneTrace desktop is implemented with HTML, CSS and JavaScript using the JQuery and JQueryUI¹ libraries and the TinyMCE² plugin for the WYSIWYG editor. The back-end is implemented with Node.js³. Steps are logged on the server as a JSON file. The Steps JSON file stored in the OneTrace server can be instantiated beforehand to make it easier to set up the environment according to potential study scenarios.

Applications automatically log Steps when a user interacts with them. Each Step is represented by a plain JavaScript object in the OneTrace server according to the Step structure defined in Figure 20. The server then generates a unique id and pushes a new Step. The list of steps is stored in JSON by the server and can be accessed and modified by the clients of OneTrace. Since it was not my primary interest to study history granularity, the OneTrace applications log Steps at a fairly coarse level of granularity:

¹ <https://jquery.com>, <https://jqueryui.com>

² <https://www.tiny.cloud>

³ <https://nodejs.org>

THE WYSIWYG EDITOR pushes a new step for each line the user adds to the text document. The source is the document's name (<docname>), the information raw data is the description of the added line (added line : <text of the line>), the author is the current user, and the temporality is the current date and time;

THE MAIL CLIENT pushes a new step for each email being sent. The source is the mail application (mailapp), the information raw data is the description of the sent email (sent mail : <text of the email>), the author is the current user, the distribution is the list of the recipients, and the temporality is the current date and time;

THE MESSAGING APP pushes a new step for each message being sent. The source is the messenger application (messengerapp), the information raw data is the description of the sent message (sent message : <text of the message>), the author is the current user, the distribution is the list of the recipients or the channel name, and the temporality is the current date and time;

4.3 CONCLUSION

In this chapter, I compiled various insights from the literature and my own empirical study to describe the Step structure, a unified history data structure. The purpose of Steps is to offer a paradigm shift from historical data being locked inside their source application to more application-agnostic history data. These data would be interpreted and represented by history artifacts, built by history tools, and manipulated or shaped by users thanks to history instruments.

I then introduced OneTrace, an example of how to implement a cross-application interaction history system using the Step unifying interaction history data structure. In OneTrace, any user actions performed on any application in the OneTrace Desktop are logged as Steps that can be uniformly manipulated by dedicated instruments. The OneTrace Desktop application also offers a simple user interface simulating a desktop environment, offering a backbone in which I can easily add applications and interaction history instruments.

In the next chapter, I explore and experiment with OneTrace to design interaction history instruments that targets specific users' needs as identified earlier in chapter 3. I want to show the design alternatives that OneTrace and the Steps offer, and how it could support users in creating application-agnostic artifacts that would increase event recall, but also coordination in collaborative settings. To do so, I also want to assess users opinions about such systems by running design workshops and user studies.

EXPERIMENTING WITH CROSS-APPLICATION HISTORY INSTRUMENTS

In the previous chapter, and based on my findings through a review of related work and an empirical study, I described the Step structure, a unified history data structure. With Steps, my goal is to find a common ground amongst history data and unify them. This should allow sharing them and manipulating them beyond their source applications and among different collaborators in one's work environment. I then described OneTrace, a proof-of-concept cross-application history system for managing Steps. I implemented a first version of this system with the OneTrace Desktop application, and detailed how applications and instruments in this environment log and interact with Steps.

In this chapter, I want to show how a concept such as Step supports the design of interaction history tools and how users would benefit from such cross-application history systems. I define two design goals in order to implement two different instruments to manipulate Steps: enhancing navigation in interaction histories with TraceLens, and enhancing re-contextualisation and better communicate and understand histories with TracePicker. Each of these instruments were designed following a participatory design process, and implemented in the OneTrace system.

To assess these two tools and the validity of my approach with OneTrace, I conducted two scenario-based observation studies in which I made participants use the tools following predetermined scenarios, and interviewed them about their experience. These two studies eventually provide evidence that a unified history can improve awareness of the timeline in a collaborative project, help users appropriate history data and generate new meaningful artifacts that enhance coordination.

5.1 DESIGN GOALS AND RESEARCH QUESTION

OneTrace and the OneTrace desktop provide an infrastructure on top of which I can design and implement specific tools to interact with history data. Based on the insights gathered from the literature and my first study, I want to design new tools that would explore how to support the practices, or to overcome specific issues that I identified. I chose to focus first on the issues related to the volume of traces generated by collaborative activities and the unified history of OneTrace. I want to design a tool allowing users to browse cross-application histories without adding too much complexity to their tasks. I also chose to support the two main practices towards histories that I identified. I want to support users in creating meaningful groups of history data, but also help them in differentiating them from one another, reshaping them in unique customized artifacts.

In other words, I want to explore designing interaction history tools for cross-application history systems such as OneTrace to enhance history data browsing, grouping, identification and sharing. I then want to assess through these tools how such a system could enhance making sense of past events and coordination on a shared timeline. With that in mind, I conducted two parallel projects based on OneTrace. Each of these tools are an example of how systems like OneTrace can improve history use. They each address one of the following design goals (DG):

- DG1 Support cross-application history browsing to improve global history awareness and coordination in collaborative environments;
- DG2 Help users express relationships between history data from different sources to enhance history data contextualization.

In order to assess OneTrace and the two future tools, I chose to conduct two interview- and observation-based user studies, following scenarios based on the interviews from my first study. I conducted one study for each of the two tools I implemented, to help answer the following research question:

- RQ3: What are the users' perceived benefits of using a cross-application history system such as OneTrace for their collaborative writing tasks?

5.2 TRACELENS : AN INSTRUMENT TO NAVIGATE CROSS-APPLICATION HISTORIES

In this first project, I wanted to address the first design goal (DG1) and find design ideas to better navigate cross-application histories and improve history data manipulation. In order to explore such design ideas enabled by the cross-application nature of OneTrace, I conducted a brainstorming workshop to generate ideas on how to make traces from different applications make sense together, and how to better interact with them, understand them, summarize them or share them.

PARTICIPANTS. One researcher involved in the study planning participated in the brainstorming session. Five additional participants were recruited, for a total of six participants (5 men and 1 woman; aged 20-30; 5 HCI Ph.D. students, 1 research engineer). Each participant had a recent experience with collaborative work and histories.

SETUP. For the first part of the brainstorming, the participants were generating ideas alone. Then they were grouped into two groups of three people for the second part. They had at their disposal various low-fidelity prototyping materials such as coloured paper, pens and pencils, rubber, post-it, transparent paper and rulers. They could also have access to tablets to shoot parts of their prototypes, and build quick video prototypes with the HCI VideoClipper¹ tablet application.

PROCEDURE. To start the session, participants were introduced to the collaborative writing applications and traces ecology as defined in chapter 3. I then introduced them to the challenges I identified in my first study and to the goal of exploring cross-application histories interactive solutions. I presented them with the two following questions to generate ideas about :

- How would traces from all your applications coexist in the same history?
- How would you be able to interact with traces to better understand them, summarize them, or share them with others?

The session was organized as follows. First, each participant had 15 minutes to generate ideas answering either or both questions, whether it be with interaction, tools or user interface design ideas. Then, forming groups of three participants, they had to vote for their four preferred ideas. They had 30 minutes to prototype them together, thanks to paper and video prototyping. Finally, they presented them to the rest of the group, and we discussed these ideas together.

¹ <https://apps.apple.com/us/app/hci-videoclipper/id1451698249?>

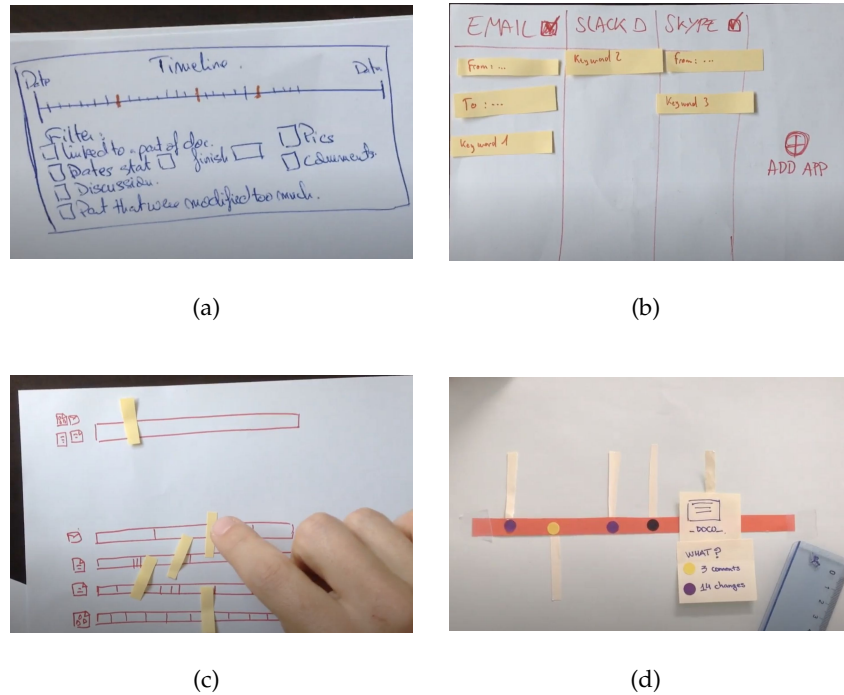


Figure 22: Examples of paper prototypes produced during the brainstorming session.

DATA COLLECTION. I collected the paper and video prototypes created by each group and participant (Figure 22). The session was video recorded.

RESULTS. Figure 22 shows four of the ideas produced by the brainstorming participants. While most of the solutions were in the form of timelines, some interesting ideas emerged. Indeed, a key issue was quickly identified: the need for new history filtering methods. In order to navigate histories efficiently, users need to be able to extract relevant pieces of the timeline as quickly as possible. Few applications enable users to filter their history, and when they do, the filters are tailored to the application. With OneTrace, history data comes from different applications and authors, which makes filtering both more necessary because of the larger amount of data and harder to realize as it has to be application-agnostic. And when compiling history data from different sources, the number of available parameters makes it hard to filter a timeline.

On Figure 22a and Figure 22b, we can see two ideas to customize a filtering command, but these designs would be way too complex and overwhelming for users. To lower the amount of data on a timeline, the idea showed on Figure 22c splits the main timeline into different independent timelines. Here, it is split according to the different

	Reification	Polymorphism	Reuse
The input: TraceLens	Reifies a filtering command as an optical lens to see the history through.	Users can use it on their workspace to filter their current timeline, on applications to filter the application's timeline, and on other TraceLens to combine them together.	TraceLens can be saved and shared among users, making the filtering command persistent and reusable.
The output: StepLine	Reifies a filtered part of history (the output of a TraceLens) as an independent timeline.	The display depends on the environment in which it appears (application, workspace..).	Users can use the content of a StepLine to create a text report, modify the content, store it and share it, reusing it as a new shared and persistent artifact.

Table 2: Instrumental interaction concepts [13] applied to TraceLens and StepLine.

sources available, such as a pre-filtering of sorts, and users can freely browse each of the timelines. Lastly, the idea showed on Figure 22d reveals another interesting idea to improve filtering. In this design, users can drag icons that represent an application, a team member, or any other object that may filter traces, and drop it onto the timeline to only show data related to the dropped object. These last two ideas give the filtering command and its output a level of direct manipulation: filters are objects you can interact with in order to combine them with a timeline in order to filter it, and filtered timelines are new timelines you can browse independently.

To support filtering in a cross-application environment such as One-Trace, I chose to keep that design idea of manipulating filters and extracted timelines as objects, and to enrich the design ideas thanks to the generative concepts [12] of instrumental interaction [9, 13]. The generative theory of instrumental interaction is defined as a conceptual lens [20] for analyzing, criticising, and enhancing the design of existing interactive artifacts. In my case, I used it to improve on the aforementioned ideas from the brainstorming session, and build upon it by exploring new ideas through the scope of the three key concepts of instrumental interaction: Reification, Polymorphism and Reuse. This led me to design the instrument that I called a TraceLens (Table 2) that reifies the concept of a filter into an object that the user can manipulate. This instrument was inspired by optical lenses: when a user looks through a TraceLens, they see a filtered part of the project's timeline as a new history artifact called a StepLine (Figure 23). With a TraceLens, users can extract the subset of history data that matches the filter represented by the lens and they can merge lenses together to create new filters. TraceLens lenses and StepLines are persistent and can be shared across users. Moreover, the reuse

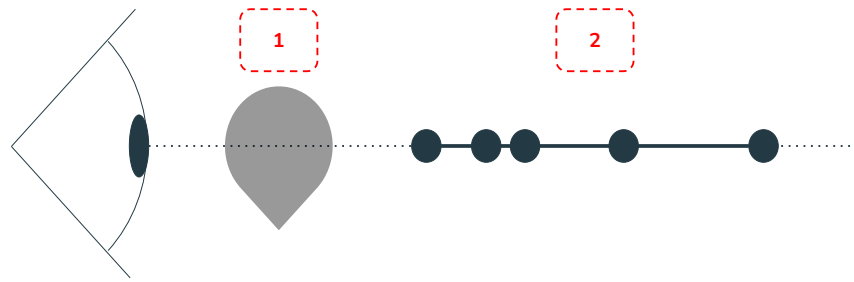


Figure 23: The TraceLens concept : The user sees through a TraceLens (1) to extract a filtered Stepline (2)

part of the StepLine allows for re-documentation by consolidating the output of a filtering command as a new editable object.

I implemented the TraceLens filtering instrument (Figure 24) in the OneTrace desktop. It features a pie-menu to browse the history as a whole (the hourglass icon), or to create lenses to filter history by application, author and tags. These TraceLens can be dragged-and-dropped onto the workspace and combined/merged by drag-and-dropping them onto each other. Each of these actions creates a StepLine, a subset of the timeline filtered by the TraceLens values. To avoid timelines with multiple branches, I chose the combination of TraceLens values to mean an “*or*” query when values are from the same group of filters (authors, applications and tags) and an “*and*” query when values are from different groups of filters, e. g. a TraceLens with the filters [mail, document, mark] performs a filtering query [[mail or document] and mark] to populate the resulting StepLine.

A TraceLens can be dragged onto an application to attach a StepLine to it, which will combine the current TraceLens output with the one corresponding to this application. If a Step belonging to an attached StepLine is magnified, it highlights the associated content in the application. It is also possible to drag-and-drop a combined TraceLens onto the application dock to save and to export a StepLine’s content as a text document composed of the content of each of its Steps. Even if it’s not currently supported, the content of the application dock is also meant to be shared synchronously among clients.

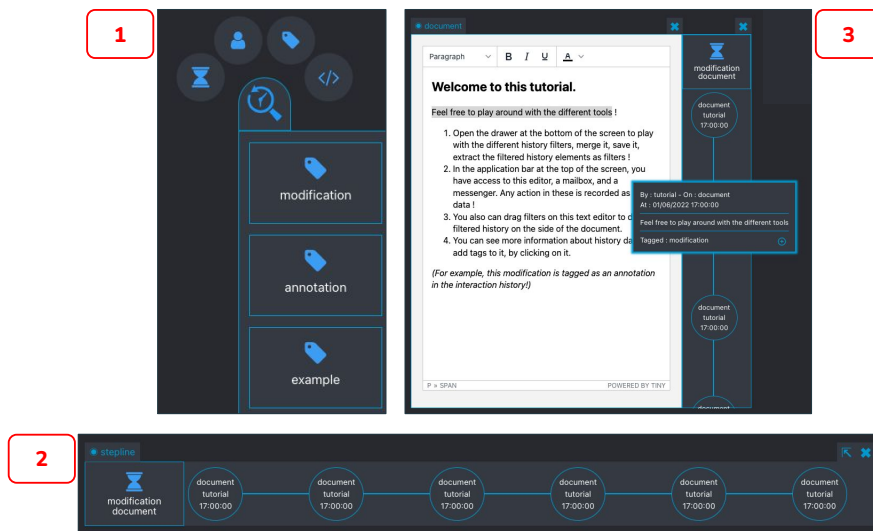


Figure 24: The *TraceLens* prototype features (1) a pie-menu to browse the available *TraceLens* and their different values; (2) Dropping a *TraceLens* in the workspace and/or combining it creates a *StepLine*, a subset of the main timeline; (3) Dropping a *TraceLens* on an application attaches a *StepLine* to it, magnifying a *Step* in this *StepLine* highlights the related content in the application.

5.3 STUDY 2 : ASSESSING TRACELENS BENEFITS

I designed *TraceLens* to help users navigate cross-application histories by creating their own customized filtered timelines. They should be able to consolidate and communicate their past activities by re-documenting histories, but also to share their history usage by sharing their own instance of *TraceLens*. This design should eventually improve their awareness and understanding of their team's past activities. With the following study, I want to assess whether users feel an improvement in navigating histories with *TraceLens* compared to their current practices (study question Q1); whether they think it would be easier to communicate their histories of a collaborative project with *TraceLens* and *OneTrace* compared to their current practices (Q2); and whether they feel that *TraceLens* and *OneTrace* improve their awareness and understanding of histories in collaborative projects compared to their current practices (Q3). To answer these three questions, I chose to conduct an observation- and interview-based user study.

I did not conduct traditional comparative studies for two reasons. First, to my knowledge, no existing system provides a flexible cross-application history system comparable to *OneTrace*, making it dif-

difficult to find a meaningful baseline. Second, using real-world applications as a baseline would be unfair because our prototype OneTrace Desktop features applications with limited functionality. Instead, each of the systems is evaluated with respect to my research question following several components that give the opportunity for us to gather qualitative data about participant's perception of such systems. It allows to compare OneTrace with their current tools, which are highly likely to be different from one another based on my first study.

PARTICIPANTS. I recruited 15 participants (10 men, 5 women; 11 age 18-25, 4 age 25-30). All were knowledge workers who had a recent collaborative writing experience. 5 of them were Master's students currently writing their Master's thesis and sharing it with their tutors. The 10 others were Ph.D. students currently writing a research article with other researchers.

SETUP. Participants used the OneTrace Desktop system in a controlled setup, in our lab, on a computer I provided.

PROCEDURE. First, I interviewed the participants about their usage of historical data, grounding it in a recent collaborative writing project. Then, I asked them to fill out a first five-point Likert-scale questionnaire, with a series of questions to evaluate their issues with the systems they currently use² After the questionnaire, I introduced them to OneTrace Desktop and the TraceLens prototype system with a short tutorial walking them through the features of the system. After this tutorial, they followed a scenario in which they had to perform a series of tasks that represent different use cases of my history tools. I prepared historical data for each task by creating a log file filled with steps to match the state of the project corresponding to each task. After going through the scenario, they filled out a second five-point Likert-scale questionnaire to evaluate how OneTrace and the TraceLens would help them with their history usage. Each session lasted about 60 minutes.

SCENARIO. In the scenario used for this study, the participant takes on the role of a person organizing their holidays with two of their friends. They use a collaborative text document as an organizational aid to consolidate the decisions they collectively make by discussing over e-mail or instant messaging and through annotations. They also have to access history data from an application that they do not have access to from one of their teammate's environment. These historical data are a browsing history of an unknown map application.

² The questionnaires are available in Appendix [A.2](#).

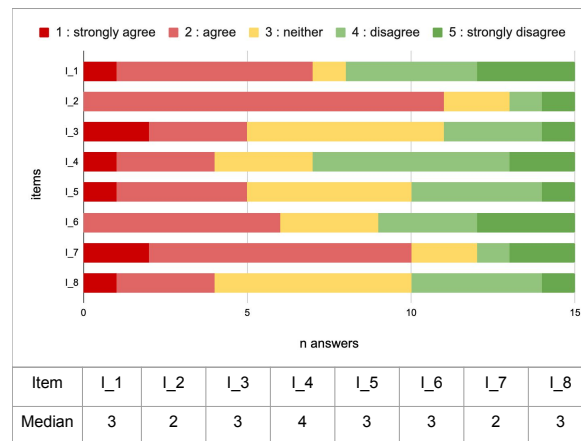
Study Questions	Questionnaire Items
Q1: Navigation	I1: Identifying traces I2: Regrouping traces I3: Navigating histories
Q2: Communication	I4: Communicating progress I5: Sharing traces
Q3: Awareness and Understanding	I6: Understanding progress I7: Recalling past actions with histories I8: Synchronizing progress

Table 3: Questionnaire items related to the TraceLens study questions.

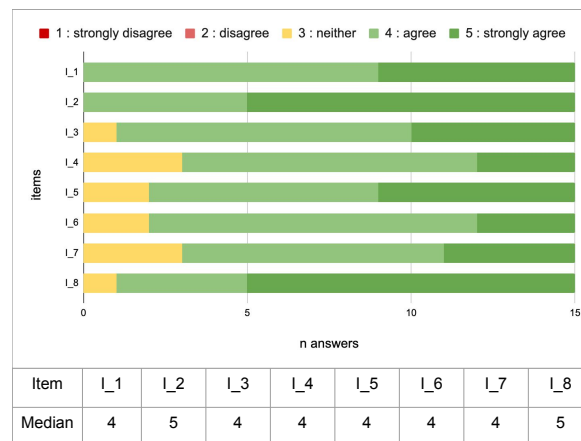
For the first task, they have to *“plan the first steps of the organization, looking back at the first discussions they had about budget, dates, and other elements they have to discover”*. Using TraceLens, they have to browse the cross-application history, build their own filters and extract relevant data from the global timeline. Then, they have to find a way to share it with their friends and communicate about the next decisions to make. The history I created for this task includes 36 Steps. For the second task, the participant *“has been busy for a couple of days and needs to get up to date with what their friends decided in their absence”*. To perform this task, the participant has access to TraceLens artifacts that their friends created and shared to understand their decisions, and can also use TraceLens themselves to navigate a larger history data set. The history I created for this task features 105 Steps.

Both of these tasks are designed to encourage the participant to use TraceLens to navigate a cross-application history, to manipulate it to organize a collaborative activity, to communicate about the different things to do with their collaborators and to understand their teammates’ past actions and decisions.

QUESTIONNAIRES. Both questionnaires feature 5 point Likert-scales (1 = totally disagree; 5 = totally agree) composed of eight items related to the questions listed earlier and detailed in Table 3. To answer the first study question (Q1), I ask participants questions about how they achieve navigating through their histories, whether for identifying relevant traces (Item 1/I1), regrouping traces when needed to better navigate it (I2), or simply browsing them as a whole (I3). To answer Q2, I ask participants if they feel that histories allow them to efficiently communicate their progress to the team (I4), or if they feel supported in sharing important traces (I5). Lastly, to answer Q3, I ask if participants think that histories are enough to understand the progress their team made on a shared project (I6), if they are able to recall past actions with the information contained in these histories (I7) and if they think it allows them to better synchronize with their team while collaborating (I8).



(a) pre-questionnaire results



(b) post-questionnaire results

Figure 25: Answers to the questionnaires of the TraceLens study. The pre-questionnaire (a) is about participant's opinion of their existing tools (oriented towards their issues, hence the reversed scale), while the post-questionnaire (b) is about their opinion on the improvements brought by TraceLens.

In the pre-questionnaires, items are statements about the participant's issues with their current tools. For example (I₂-Pre): *"I happen to face issues in regrouping traces left within different histories, but conveying the same kind of information"*. In the post-questionnaire, items are the same statements, but oriented towards the potential benefits of OneTrace and TraceLens. For example (I₂-Post): *"This kind of system would improve regrouping traces left within different histories, but conveying similar information"*.

DATA COLLECTION. I took notes during the interviews and collected the answers to the two questionnaires. Participants were encouraged to think aloud while answering the questionnaires and while reflecting on their actions when performing the tasks. Each session was video recorded.

ANALYSIS METHOD. I used a mix of qualitative and quantitative methods. Questionnaire data were analyzed with summary statistics. Think-aloud data were annotated by the researcher conducting the experiment, focusing on feedback and opinions about OneTrace and cross-application history systems. Videos were transcribed by the researcher in charge of the experiment.

5.4 RESULTS AND DISCUSSION

Figure 25 shows the answers to the pre- and post-questionnaire as well as the median rating for each of the items. The median rating for each question increases between the pre- and post-questionnaire, except for question I_4 , where it stays the same. Given the small number of participants ($n = 15$), a significance test is not robust. Nevertheless, we report the results of the Mann–Whitney U test (or Wilcoxon rank-sum test), which is significant for all questions except I_4 ($p = .141$).

The increase in median rating for items I_1 ($3 \rightarrow 4$, $U = 54$, $p = .016$), I_2 ($2 \rightarrow 5$, $U = 12.5$, $p < .00001$) and I_3 ($3 \rightarrow 4$, $U = 32$, $p = .001$) suggests that OneTrace and the TraceLens helped participants navigating their histories, identifying relevant traces and grouping traces across application (Q_1). About median ratings for communication items, I_4 ($4 \rightarrow 4$, $U = 76.5$, $p = .141$) is stable while I_5 ($3 \rightarrow 4$, $U = 39$, $p = .002$) shows a slight increase. This is still important to note that while the median values are somewhat stable, the post-questionnaire only shows neutral to positive (3 to 5) ratings, while the pre-questionnaire also shows negative ratings (1 to 2). While it would need further investigation, these results suggest a light improvement for participants to communicate their progress or share important history data with their teammates (Q_2). The median ratings for the items all show an increase for understanding the global progress of a collaborative project (I_6 , $3 \rightarrow 4$, $U = 64.5$, $p = .049$), recalling past events thanks to histories (I_7 , $2 \rightarrow 4$, $U = 36$, $p = .002$) and synchronizing progress with other team members (I_8 , $3 \rightarrow 5$, $U = 25$, $p = .0003$). These results suggest that participants felt that TraceLens could improve their awareness of the timeline and their understanding of their team’s progress and past actions (Q_3).

In summary, all ratings but one increased substantially between the pre- and post-questionnaire, suggesting that participants found OneTrace and TraceLens to be useful systems to better navigate histories (Q_1), share progress on collaborative activities (Q_2) and have a better awareness of every other team member’s progress (Q_3). This increase was not significant enough on one of the items, and needs further investigation.

Most participants (12 out of 15) expressed their interest in manipulating filters as objects with TraceLens to create their own filters, consolidate them and share them for later reuse. They also liked the OneTrace main timeline's feature for adding "tags" on some of the Steps. This allowed them to add meaning to the Steps and create another level of information for categorization, thus helping them communicate meaning through histories. Participants also found that tagging really added to the filtering freedom enabled by TraceLens, allowing them to quickly create a custom filter to find or share relevant Steps. However, three of them expressed concerns about projects with larger team sizes or longer duration, which could lead to too many tags which meaning may be lost. The main alternative feature for navigating a filtered timeline was to consolidate its content as an editable, shared text document. While ten participants used this feature extensively to navigate timelines with many elements, the others were satisfied with a filtered timeline and went through the elements one by one. However, the technique they used to understand past actions on the project seemed to have no influence on their ability to understand the project timeline.

Overall, the study results showed that the TraceLens instrument seemed to achieve its goal (DG₁) of supporting cross-application history browsing, sharing and awareness. Combined with the OneTrace system, participants found improvements in navigating histories (Q₁), even when considering the increase in information due to the merging of histories from different sources. Participants also found improvements in both understanding the timeline of events (Q₃) and sharing their own progress through it (Q₂).

My observations showed that while OneTrace implements a novel approach by having a single history system for the entire collaborative work context, participants felt comfortable with the system and had a good understanding of both the main timeline and the TraceLens instrument. Five participants explicitly expressed that cross-application history would be a real "game-changer" for them in terms of collaborative work. Participants quickly understood how to use TraceLens to find information in different ways, e.g. to look at the timelines or at a previously created TraceLens, and when looking to group and consolidate information by exporting StepLines as text documents. While three participants over-used the "export-as-text" feature for many navigation tasks, five participants did not like it at all and preferred navigating the timeline element by element. I also noticed that once a StepLine was created, there was a confusion between it and the TraceLens object that generated it. Because the two objects are not different enough, some participants did not understand that they could navigate the generated StepLine as a regular timeline. Future work should improve the design to make the distinction clearer.

With respect to RQ₃, overall participants found OneTrace and TraceLens beneficial in terms of coordination and timeline awareness in collaboration. The study also shows that a concept such as Steps allows to implement simple yet powerful instruments such as TraceLens to address the increased amount of information in histories and to enhance browsing and manipulating history data.

5.5 TRACEPICKER : AN INSTRUMENT TO ENHANCE HISTORY DATA CONTEXTUALIZATION

In this second project, I wanted to explore design ideas related to the contextualization of history data and address the second design goal (DG₂). To this end, I wanted to understand first which criteria in histories help users understand the context of past actions, to understand why an action has been performed and how building relationships between traces might help. I also wanted to know how users would communicate it to other members of their team, and what would make it easier to understand a timeline of past actions to get the current state of a project. This second project was entirely executed with the collaboration of Kevin Ratovo, who was an intern in the research team at the time of the project.

I chose to run a participatory design workshop on how to help users re-organize history data to resume work after a break, understand the work of a teammate and join a collaborative project. This workshop took place over three participatory design sessions.

PARTICIPANTS. One researcher in charge of the study participated in each of the three sessions. Three additional participants were recruited for each session for a total of 9 external participants (6 men and 3 women; aged 20-30; 2 software engineers, 3 HCI Ph.D. students, 1 research engineer, 3 HCI Master's students). Each participant had a recent experience with collaborative work (Master's thesis, research article or technical report).

SETUP. For each session, the four participants were split into two groups of two. We provided each group with the same set of traces left across multiple histories during a collaborative project. They had access to a collaborative document, a set of historical data from the actions performed on and around this document, and a scenario explaining the collaborative project. The traces were paper Post-It notes describing a user interaction, with information corresponding to the attributes of a Step. Each Post-It note informed about the provenance and type of trace (an annotation, a comment, a modification, or exchanged messages), the content of the trace, the time when the trace was left and the author of the trace, for a total of thirty-two Post-Its.

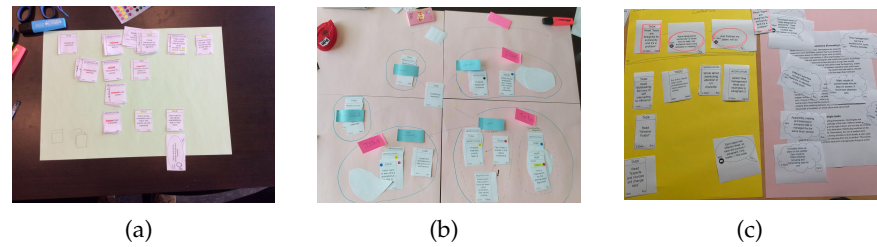


Figure 26: Examples of history artifacts and groupings produced during the participatory design workshop. Each grouping is a sequence of traces. In (a) these sequences can be collapsed under the most important trace of the sequence; in (b) these sequences are themselves grouped into abstract tasks; in (c) these sequences are grouped by spatial context.

PROCEDURE. Participants had to create groupings of these traces that made sense to them and that they could communicate to the other group. First, we asked them to create one or more paper-based artifacts by regrouping the historical data. The groupings had to be meaningful for situations such as resuming work after a long break. They also had to communicate the right information to other people, e.g. to describe progress to their co-workers. The only explicit limitation was that they had to challenge current tools and representations by trying to avoid purely temporal structures such as timelines. We then asked each group to present their work to the other group to see if they understood each other’s historical data groupings, and then discuss the criteria they used to create these groupings.

DATA COLLECTION. We collected the artifacts created by each group (Figure 26). Each session was video recorded.

RESULTS. Figure 26 shows some examples of the artifacts produced by the participants of this workshop. To express relationships between history data, participants used different representations such as blocks of traces with common attributes, but also with links or lines between traces, which were not necessarily oriented. For example, they mainly created groupings that expressed sequences of events and causality (Figure 26a) between the different events that occurred. They also made groupings based on abstract tasks that the user was dealing with at the time (Figure 26b), or related to what the user had in mind when doing specific actions. In the last example shown in Figure 26c, traces are grouped by the spatial context of the main shared document, to show the chain of events that led to specific modifications in the document even if those traces are from external tools such as communication channels.

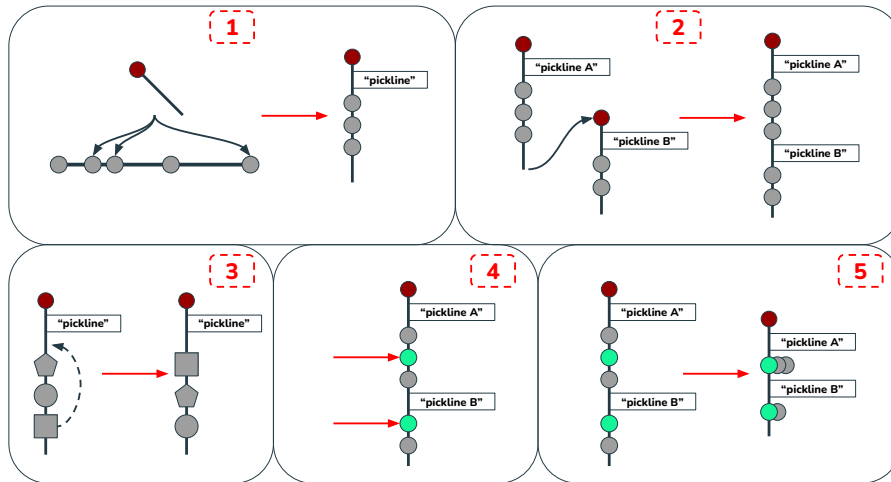


Figure 27: The TracePicker concept: (1) The TracePicker instrument is used to pick steps and pin them together, creating a PickLine; (2) PickLines can be joined to create sub-groups of steps; (3) the content of a PickLine can be re-organized; (4) steps can be set as important; and (5) PickLines can be collapsed and expanded.

In all of these cases, participants needed ways to express what their thoughts were behind a specific group of traces, e.g. by labeling these groups or linking them to other content. They also wanted to remain in control of what was in these groups and did not mention any automated groupings. Participants also mentioned that some traces are more important than others to understand the current state of the shared document. They expressed this notion of importance in their groupings by having a “collapsed” representation of the group of traces, showing only important history data. Then, an “expanded” version of these groupings allows to dig into the details of the different actions performed in that specific context. An example of this idea can be seen in Figure 26a. Finally, participants emphasized that groupings may differ depending on the team members they would communicate them to, so as to give different contextual information depending on the audience.

Based on the results of this workshop, I designed TracePicker (Figure 27), an instrument that lets users re-document traces in a fully customized way. With this instrument, users can pick steps and pin them together for expressing meaningful relationships between them. Reflecting on the concepts of Causality and Priority that I introduced in the Step structure, TracePicker lets users create custom clusters of history data to organize them in a non-temporal way. The goal of this

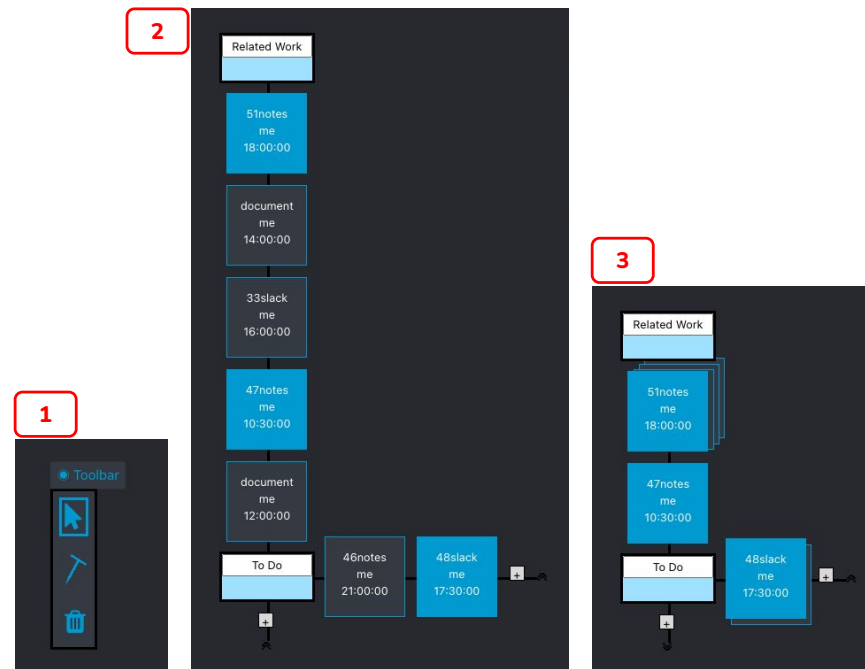


Figure 28: The *TracePicker* prototype features (1) a toolbar to create and delete *TracePicker* objects on the workspace. (2) Users can pin Steps and other *PickLines* with a *TracePicker* to form a new *PickLine*, and (3) shrink a *PickLine* to display only the important Steps.

instrument is to allow users to express the context in which past actions occurred by grouping traces together, and better understand or communicate the current state of the project through interaction history data. They can also express a hierarchy between steps by setting some of them as more important than others, allowing them to “collapse” a *PickLine* to only make the important steps and the groups’ labels visible, and put the other steps in the background for synthesis and clarity purposes.

I implemented the *TracePicker* instrument based on the *OneTrace* desktop interface and logging system previously implemented (Figure 28). It features a toolbar to create *TracePicker* objects on the workspace and to delete elements in the *PickLines*. Each *TracePicker* has a label that users can modify. A *TracePicker* can be used to pick-and-pin Steps from the timeline to form a *PickLine*, or to pick other *PickLines* already in the workspace to form a sub-*PickLine*. Users can further organize the elements within a *PickLine* by drag-and-dropping them. Double clicking a Step from the main timeline or from the *PickLine* itself toggles it as important. It is possible to collapse the *PickLines* to only show the important Steps. Drag-and-dropping a *PickLine* on the application dock saves it so that it can be accessed and reused by all team members.

5.6 STUDY 3 : ASSESSING TRACEPICKER BENEFITS

I designed TracePicker to give users a way to create new history artifacts by building custom groups of history data and linking them together across applications rather than being constrained by the temporal ordering of a timeline. The goal is to help users contextualize history data by letting them build new artifacts for themselves or for their team. By allowing users to reshape these artifacts into a collapsed form, the goal is also to reduce the amount of data that users have to browse through in order to understand the current state of the project. Specifically, I want to know whether users think TracePicker could help them reduce history bloat and focus on important traces compared to their current practices (study question Q4); whether users think TracePicker would help them better contextualize traces left around their project compared to their current practices (Q5); and whether users feel they would be able to resume work more easily with TracePicker compared to their current practices (Q6). To answer these three questions, I conducted an observation- and interview-based user study, in the same way I did for the TraceLens study.

PARTICIPANTS. I recruited 8 participants (7 men and 1 woman; aged 20-25) with a recent collaborative writing experience. 6 participants were HCI Master's student currently writing their Master's thesis and sharing it with other people including their tutors. The 2 other participants were Ph.D. students currently writing a research article with other researchers.

SETUP Participants used the OneTrace Desktop system in a controlled setup, in our lab, on a computer I provided.

PROCEDURE. First, I interviewed the participants about their usage of historical data, grounding it in a recent collaborative writing project. I then asked them to fill out a five-point Likert-scale pre-study questionnaire, with a series of questions to evaluate their issues with the systems they currently use³ Then I introduced them to the OneTrace Desktop and the TracePicker tool with a short tutorial, walking them through the system's features. After this tutorial, I asked them to follow a scenario (see below) where they had to perform a series of tasks representing different use cases. I prepared historical data for each task by creating a log file filled with steps to match the state of the project corresponding to each task. After going through the scenario, I asked them to fill out a five-point Likert-scale post-study questionnaire to evaluate how OneTrace would help them with their history usage. Each session lasted about 60 minutes.

³ The questionnaires are available in Appendix A.3.

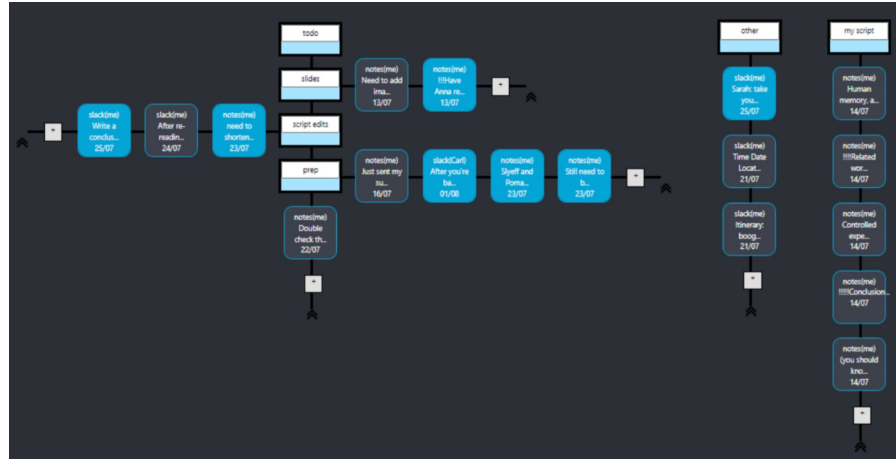


Figure 29: The PickLines provided for the second task of the study.

SCENARIO. Participants impersonate Sarah, a Master’s student currently writing her internship report. For the first task, *her two supervisors, Anna and Carl, have both recently given her feedback on her draft, while she also has her own ideas on content to add or modify. Her workday is ending, and she wants to compile information about what she has to do tomorrow.* To perform this task, the participant can freely read the timeline and the traces, and open the applications in the OneTrace Desktop. Then, they are expected to create one or more PickLines with the TracePicker tool to help them better manage their return to work the next day. The history I created for this task includes 53 Steps. For the second task, *Sarah just returned from 2 weeks of vacation. She prepared multiple PickLines to remember better what she did and what she had to do by the end of her internship.* To perform this task, the participant has to answer a list of questions provided by the examiner about the different things Sarah has to do after returning from her break, using pre-built PickLines to find the requested information. Three PickLines were provided for this task: one with three Steps, one with five Steps and a more complex one with ten Steps and three branches (Figure 29). These two tasks aim to see how participants create a PickLine and arrange Steps on it, and how they use existing PickLines to understand and contextualize past actions.

QUESTIONNAIRES. Both questionnaires feature 5 point Likert-scales (1 = totally disagree; 5 = totally agree) composed of seven items related to the study questions listed above and detailed in Table 4. To answer the first study question (Q_4), I ask participants if they feel in control over what traces are displayed and available to them in the histories (Item 1/ I_1) and if they feel able to find relevant traces in a large amount of information (I_2). To answer Q_5 , I ask participants if they feel able to contextualize their own past actions (I_3) and their

Study Questions	Questionnaire Items
Q4: Focus on important traces	I1: Control over displayed traces I2: Dealing with a lot of information
Q5: Contextualize past actions	I3: Contextualizing their own traces I4: Contextualizing their team's traces
Q6: Resume work	I5: After a short break I6: After a long break I7: In general

Table 4: Questionnaire items related to the TracePicker study questions.

teammates' traces (*I4*). Lastly, to answer *Q6*, I ask participants if they feel able to resume work with the tools at their disposal after a short break (*I5*), a long break (*I6*) and in general (*I7*).

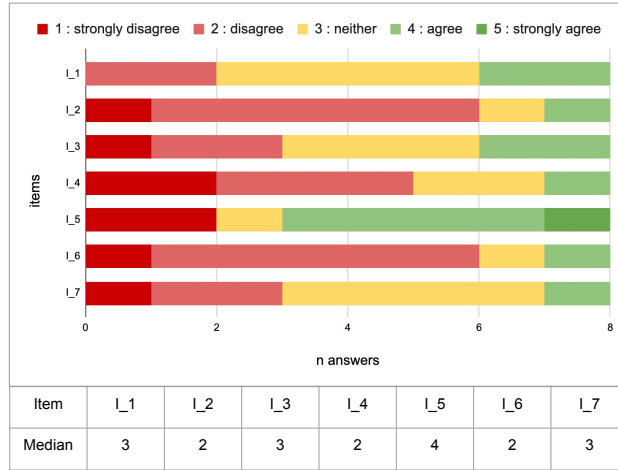
In the pre-questionnaire, the statements are about the participant's use of current tools. For example (Item₃/Pre-questionnaire): *"With my current tools' histories, it is easy to contextualize my own past actions"*. In the post-questionnaire, the statements are the same but about the potential benefits of OneTrace and TracePicker. For example (Item₃/Post-questionnaire): *"With this system, it would be easier to contextualize my own past actions"*.

DATA COLLECTION. I took notes during the interview and collected the answers to the two questionnaires. Participants were encouraged to think aloud while answering the questionnaires and while reflecting on their actions when performing the tasks. Each session was video recorded.

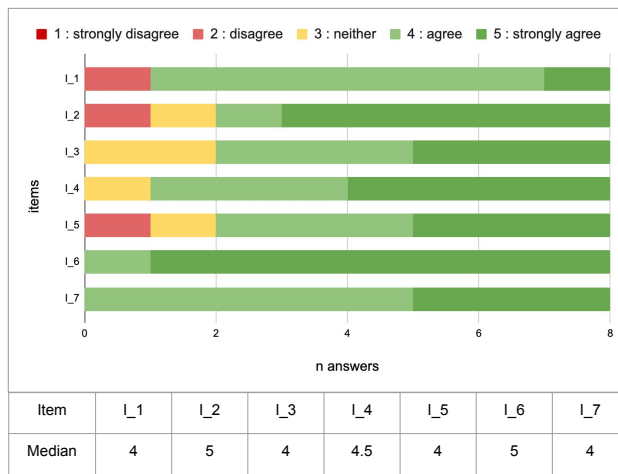
5.7 RESULTS AND DISCUSSION

Figure 30 shows the answers to the pre- and post-questionnaire as well as the median rating for each of the items. The median rating for each question increases between the pre- and post-questionnaire, except for question *I5*, where it stays the same. Given the small number of participants ($n = 8$), a significance test is not robust. Nevertheless, we report the results of the Mann-Whitney *U* test (or Wilcoxon rank-sum test), which is significant for all questions except *I1* ($p = .052$) and *I5* ($p = .317$).

The increase in median rating for items *I1* ($3 \rightarrow 4$, $U = 13$, $p = .052$) and *I2* ($2 \rightarrow 5$, $U = 6.5$, $p = .009$) suggests that TracePicker helped participants focus on important historical data and reduce data bloat (*Q1*). Participants appreciated that they could mark some Steps as important and only display the most relevant ones. However, they would have liked the ability to add other information, such as color



(a) pre-questionnaire results



(b) post-questionnaire results

Figure 30: Answers to the questionnaires of the TracePicker study. The pre-questionnaire (a) is about participant's opinion of their existing tools, while the post-questionnaire (b) is about their opinion on the improvements brought by TracePicker.

coding, and archive historical data that is unlikely to be used again. Similarly, the increase in median ratings for the contextualization items I_3 ($3 \rightarrow 4$, $U = 10$, $p = .024$) and I_4 ($2 \rightarrow 4.5$, $U = 3.5$, $p = .003$) show that TracePicker seems to make it easier for participants to express and understand the context of past actions in an environment with multiple shared artifacts (Q_2). Regarding the items about resuming work, the median rating for resuming after a short break is stable (I_5 , $4 \rightarrow 4$, $U = 22$, $p = .317$) while it shows an increase (I_6 : $2 \rightarrow 5$, $U = 0.5$, $p = .001$; I_7 : $3 \rightarrow 4$, $U = 2.5$, $p = .002$) for resuming after a medium or long break. These results suggest that TracePicker improves resuming work except for short breaks (Q_3). We assume that participants rely on their memory rather than their tools after a short

break. Indeed, two participants said that they do not have *“issues remembering things after a short break”*.

In summary, all ratings but one increased substantially between the pre- and post-questionnaire, suggesting that participants valued the features that put them in control of historical data to help them better understand the context of past actions. We created the TracePicker instrument to enhance historical data contextualization by letting users create custom groupings of Steps. Study results suggest that the PickLines created with TracePicker make it easier to contextualize past actions (Q5), resume work after a medium or long break (Q6), and reduce bloat by focusing on important historical data rather than details (Q4). When resuming work after short breaks, participants did not see the benefit of organizing their history rather than simply relying on their memory and saw it as a waste of time.

Participants liked the ability to create meaningful new artifacts from their history, because *“you can just put everything you need there and refer to it later”*. However, when resuming work after short breaks, they did not see the benefit of spending time organizing their history rather than simply relying on their memory. Participants also appreciated the way they could mark some Steps as important to then be able to only display the most relevant ones. But they would have liked to add other custom layers of information, e.g. by adding an *“archiving”* feature to archive history data that would unlikely be used again in the future. Giving a trace such a status of archived trace could be achieved by adding a third lower level of importance. They would also like to *“color code”* the important traces on top of the labeling feature as an additional visual cue.

Additionally, I identified two different patterns when observing participants performing the second task related to understanding past actions using a provided PickLine. The first pattern refers to participants who judged that the structure of the PickLines, their labels and the way they were grouped, was enough to get the needed contextual information. A second pattern was from participants who, on top of the PickLines, also browsed the shared document and the main timeline, mostly to be sure not to miss important information. While these participants did not explicitly mention the lack of interactivity between a PickLine and related content, I believe it would be a clear improvement to support it in future designs in order to further improve contextualization.

Overall, the TracePicker instrument I designed seemed to achieve its goal (DG2) of enhancing history data contextualization by letting users create custom groupings of Steps. Study results suggest that the PickLines created with TracePicker make it easier to contextualize

past actions (Q5), resume work after a break (Q6), and reduce bloat by focusing on important history data rather than details (Q4).

While it seems that my system successfully addressed the issues I identified, there is room for improvements. First, users could have even more control over the creation of custom history artifacts such as PickLines, by freely tweaking and reshaping these artifacts and their content. Indeed, my current design lacks flexibility and is still “*line-shaped*”, and the content of the PickLines are purely Steps enriched by a new object encapsulating it. Second, Picklines or other future history-based artifacts could reference not only history data but also relevant parts of content from different sources to further consolidate context around history data. For example, they could reference parts of a shared document in a discussion and include these cross-applications links in histories.

With respect to RQ3, users overall found that TracePicker facilitated the contextualization of past actions. The positive feedback reinforces the relevance of cross-application history environments as offered by OneTrace to better contextualize past actions, a recurring task performed on histories, and thus improve the use of interaction histories and associated tools.

5.8 CONCLUSION

In this chapter, I defined two design goals, to implement two different instruments for manipulating traces. The goal was to experiment with Steps and OneTrace to provide users with two examples of cross-application interaction history instruments. I also wanted to show through these two designs how the cross-application of OneTrace is beneficial for interaction history use, especially in collaborative writing settings. To satisfy my first design goal (DG1), I designed TraceLens to improve interaction history use by enhancing history browsing and manipulation. The design methods included mixed methods, a brainstorming session with other design researchers, and the generative concepts of instrumental interaction. For the second design goal (DG2), I designed TracePicker, an instrument allowing users to customize groupings of traces, enhancing history data contextualization. I designed TracePicker based on the results of a participatory design activity in which participants had to create groupings of a given set of traces to make it easier to understand. These two tools were implemented in OneTrace and evaluated following a similar observation- and scenario-based user study. The results provide evidence that a unified history can improve awareness of the timeline in a collaborative project, help users appropriate history data and generate new meaningful artifacts that enhance coordination (RQ3).

As with any studies, these studies have limitations. First, the number of participants was small and the study should be replicated and extended. Second, these were short-term lab studies, so I could not assess the scalability of the approach to longer periods and larger histories. Long-term field studies should address these questions. This would require integrating OneTrace with existing tools since the limited features of the applications in our proof-of-concept implementations are unsuitable for real work over an extended period. Finally, the scenario-based approach meant that the participants had to understand the history of a project in which they did not participate beforehand. I therefore could not assess the impact of the participants' memory on their assessment of the system. On the other hand, this created a situation often encountered in collaborative settings when a new team member joins an existing project and has virtually no knowledge of the actions performed on shared content. Overall, our two instruments met the goals they were designed for, and showed the relevance of our approach with OneTrace, the Steps unifying data structure, and cross-application interaction histories in general. It shows evidence that cross-application, cross-user histories and associated tools should be an integral part of collaborative software.

These two user studies also led me to reflect on my approach with the Step unifying model for history data. Indeed, observing how people appropriate interaction history data to make something new out of it made me rethink what to consider a "common attribute", especially when it comes to how relationships between Steps are stored. If one thinks of a Step as a singular history data, it does not make sense in and by itself. But if you associate it with other Steps, thanks to an instrument, it builds links between steps, or allow users to do so. For the last part of this thesis, I want to iterate on my Step structure and emphasize the difference between interaction history data and the relationships that tie them together into meaningful objects. I want to make it easier to understand the environment that ties together history data, history data collections or artifacts, and history tools or instruments within a specific context of activity. Reflecting on works on interaction substrates and instruments [11, 12, 43], I need to clarify what are Steps and their attributes, and how history instruments may allow to build relationships among Steps.

INTERACTION HISTORY SUBSTRATES

In the previous chapter, I experimented with cross-application histories by designing two different instruments to manipulate traces outside of applications' environments. These instruments were based on OneTrace, my Unified History proof-of-concept system. They aimed at enhancing interaction history use by providing users with a flexible environment letting them use different instruments depending on their goals. To achieve such flexibility, I designed these instruments following the principles of instrumental interaction [9, 12, 13].

I assessed the impact of these instruments with two user studies in which users could create different artifacts made of interaction traces. Observing these tools in action, I can highlight the following critiques. First, based on my Steps model, it is unclear which attributes can be modified and which attributes can only be read. There is a need to define clearly what makes a historical data attribute editable by users, to avoid any confusion in historical tools design, but also to not alter the true meaning of the edited historical data. Editing historical data should augment the data and not erase information from it. The second point is about what I called the "history artifacts" that the use of history instruments creates. For example, I defined a filtered timeline created by a TraceLens as a new history artifact. But it is not well defined what exactly these artifacts are and what are their properties. This leads me to my third point, which is about relationships among Steps. In my current model, relationships among Steps are stored in the Steps data structure, technically by storing Steps IDs as pointers. This relationship can be abstract (such as temporality) and dedicated instruments can reveal them through a new artifact (such as a timeline). It can also be created by history instruments themselves by creating new artifacts (such as the sequences created by TracePicker). Consequently, I think that relationships should not be a part of the Steps model but rather the interpretation of an instrument or a user while building new history data collections that structure a history artifact. Finally, I first implemented OneTrace as an exploratory basis for cross-application history environments. Compiling the concerns mentioned above should result in a more formal description of such an environment. It needs to be clear how each piece of digital information and each component interacts with each other. OneTrace should be a robust, flexible and extensible basis for future work on cross-application interaction histories.

In this chapter, I want to address these points. It should be made clear what history data attributes are to be accessed and/or modified, in order to design history instruments that are less confusing to users. The concept of “history artifacts”, their attributes, their constraints and whether or not users are able to interact with them as they can interact with Steps also needs to be better described. The nature of the relationships among Steps should be reassessed, with the use of history instruments making it possible to interpret relationships among historical data or build new ones. Treating relationships among Steps as a construct of instruments would help generalize it. The applications generating a Step should not need to have access to the history to set an attribute that will translate into relationships when read by an instrument. For example, all Steps contain a “date” attribute, a timestamp, and the temporality relationship draws the timeline of events by sequencing these timestamps. But it should be up to the instrument building the timeline to draw this relationship. An application creating a step should not have to access the history and read all history data in it to set the value of this temporality relationship.

Finally, I want to clarify that a cross-application history system such as the one I want to define should be activity- (or context-) centric rather than document-centric. This environment definition should include how applications, history data, and history instruments all work together. To this end, I describe more precisely what would be an environment of cross-application interaction histories through the lens of the interaction substrates and instruments interaction model described by Beaudouin-Lafon [10, 11] and explored in various other works [33, 43]. The final goal is to provide a well-defined framework that would be the basis for future works on designing interaction history instruments in flexible and extensible environments. As an example, I describe a new implementation of my OneTrace proof-of-concept system based on this framework and the Webstrates environment [43].

Foreword : A Reminder of the Concepts of Substrates and Instruments.

As I introduced in Chapter 2: “Background and Context”, my main research goal in this thesis is to explore the concept of more flexible interaction histories and draws its inspiration from Beaudouin-Lafon’s related work on the model of interaction substrates and instruments [10, 11]. An *interaction substrate* represents a digital object’s affordances, it defines the constraints and properties of digital information. A single digital object can be composed of different substrates, representing different layers of information, each with their own affordances and with persistent relationships between one another. These collections of substrates represent a rich digital material to be interacted with (Figure 11). To operate on interaction substrates, Beaudouin-Lafon et al. [9, 13] introduce the principles of instrumental interaction. These principles describe an *instrument* as a first class object reifying one or several commands. These commands are polymorphic, i.e. they can operate on different types of objects, and their input and output can be accessed and reused later. Ultimately, instruments are interaction substrates that have the capabilities of acting on other substrates. A digital environment composed of interaction substrates and instruments then provides more flexible interfaces to users, allowing them to reflect on which tools can interact with which object, depending on their goals, their preferences and the affordances they perceive.

With Steps, I explored how we can free history data from their applications and provide a flexible environment that lets users manipulate it with different instruments according to their goals. The way I redefined interaction history and history tools is directly related to interaction substrates and instrumental interaction [26, 30, 33]. With the knowledge I gathered during this thesis, I can now articulate what is a History Substrate and provide a framework to design better history tools. In this chapter, in order to formalize the concepts introduced with Steps and the OneTrace system, I want to define what is a history substrate, and what set of constraints and properties defines history data in general. I also want to define what is a history instrument that operates on the affordances and constraints of Steps, and characterize the history artifacts that result in the use of such history instruments. Indeed, such history artifacts that are factually collections of Steps are containers of history elements that share similarities with the definition of substrates I introduced earlier.

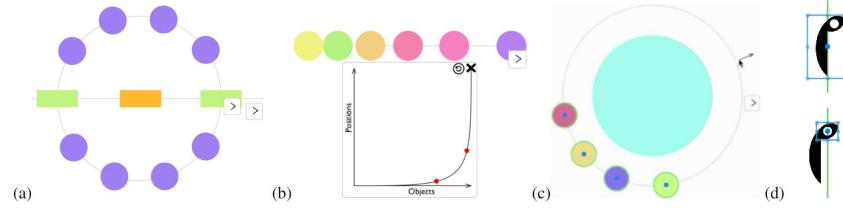


Figure 1: *StickyLines* reify alignment and distribution into first-class graphical objects that users can manipulate directly. (a) Circular and horizontal alignments. (b) Non-linear distribution. (c) Ghost guideline. (d) Tweaking an object's bounding box.

Figure 31: The *StickyLines* instrument from Felice Ciolfi et al. [23].

6.1 UNDERSTANDING HISTORY ARTIFACTS : WHAT IS A TIMELINE ?

In order to interpret history data as substrates, we need to determine the properties of an object representing past events that we can shape into affordances to interact with using dedicated instruments. To understand the objects that I called "*history artifacts*", which result from the manipulation of history data, we need to answer the following question: what is a timeline? If broken down, a timeline can be seen as both an instrument and the resulting object it produces. It orders interaction history data from a specific context in a temporal manner. This produces an artifact that makes the history data temporal hierarchy, or relationship of temporality, tangible for users. We can even argue that this relationship among history data is only an abstract one until it is made visible through the right lens.

To better explain the decomposition of history artifacts, instruments and substrates, I draw a parallel between history timelines and Felice Ciolfi et al.'s [23] *StickyLines*. Their work describes a *StickyLine* (Figure 31) as an instrument that orders graphical objects and that can help break down what is a timeline through the lens of instrumental interaction [10, 11]. A *StickyLine* is an instrument that reifies alignment and distribution commands of graphical objects into a first-class object similar to a magnetic guideline. Using this instrument on other graphical objects then applies a constraint on these other objects' properties. In the case of *StickyLine*, users can manually associate a graphical object to a *StickyLine* to apply the associated constraints on these objects' properties, such as their anchor point in a 2D space. Once associated, these constraints will dynamically apply whether the *StickyLine*'s parameters or the graphical objects' attributes change values. This case is a good illustration of the interaction substrates and instruments interaction model. Graphical objects are objects with their own attributes and properties that can be manipulated by instruments recognizing these properties. *StickyLines* are instruments that apply specific constraints on graphical objects' position in a 2D space, and they also are substrates with their own properties by reifying the

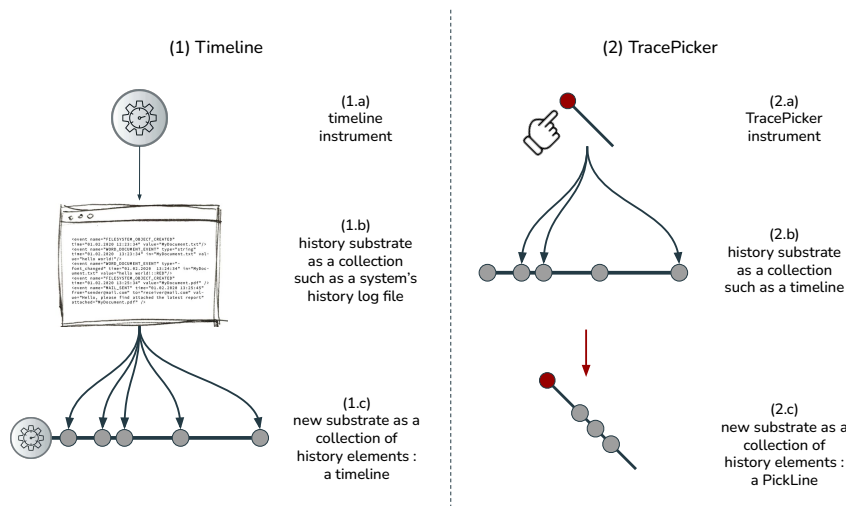


Figure 32: History instruments and how they interact on substrates to create a new substrate. Both the Timeline (1) and TracePicker (2) are instruments (a) that collect history elements within history collections (b) to create new artifacts, or history collections, as new history substrates (c), with their own properties and new relationships interpreted or created among history elements.

alignment command of one or more objects into a new graphical object. This reification of the alignment command into a new graphical object with its own position in a 2D space makes StickyLines polymorphic, and it is possible to interact on the created StickyLine with another StickyLine to apply a constraint with a new StickyLine on an already created one. Associating graphical objects with a StickyLine is also a way to create groupings of objects. By doing so, users can create new complex artifacts composed of various substrates, of graphical objects put into relationships by the common constraints applied by the instruments with which they are associated.

A timeline is similar to this type of substrate. The way it is built resembles the way a StickyLine applies a constraint to graphical objects. It is a container of elements sharing similar properties and may apply specific constraints on these properties. A timeline object applies in a given context (an application, a document, a system...) and puts together history data to create a new artifact, a grouping of history data. These history data are then history element that the timeline applies a constraint on for ordering them. The resulting artifact is a history substrate, a collection of history element put together following their temporal property. While a timeline automatically builds a history substrate following its constraints, we can also design instruments such as TracePicker that allows more direct manipulation of history elements. With TracePicker, users can collect history elements within other collections to create their own collections with their own layouts following customized constraints (Figure 32). Based on this

framework, thinking of history instruments as commands to collect history elements and create new substrates in the form of a collections allows for more interoperability among history instruments, such as filtering and extracting histories from different places with TraceLens. It also allows for more interoperability among substrates or collections, such as using TracePicker for collecting history elements from different applications and gathering them as a new substrate. History collections could also be polymorphic substrates that would have different affordances depending on the instruments used to shape their layout. For example, one could filter these collections such as an instant messaging history, or order them based on the layout of a document of interest as in text editors.

The main difference among history instruments is how they may interpret relationships, or how they allow users to build relationships among history elements within a collection. Some instruments such as a simple timeline builder or a TraceLens create a new history substrate based on a relationship between properties that can only be accessed and not modified, such as the time when the associated trace was left by a user. The relationship among history elements in that case is completely up to how the instrument reads and interprets the elements' properties. In the case of tools such as TracePicker, the created substrates put history elements in relation with each other through properties that can be accessed and modified by users themselves. It allows them to create new substrates based on what affordance they perceive from history history elements contained in the former substrate. For a collection created with TracePicker for example, users can express abstract sequences of actions, and a hierarchy of importance inside these sequences. This difference shows that relationships are not a part of history data but rather part of collections that were interpreted and built by specific instruments through specific operations.

Creating relationships among history data is an important concept because it is how users can make sense of past data. To create these relationships, we need to design history instruments that are clear about how they behave with history elements properties and constraints for creating new substrates that put different histories in relation. What this reflection on history artifacts as substrates reveal is that, while some attributes can only be read from traces, others can also be modified and tweaked. This needs to be clarified in a new iteration of the Step unifying data structure that I introduced earlier in this thesis.

6.2 FROM STEPS TO HISTORY SUBSTRATES : ITERATION ON THE STEPS MODEL

With the Steps unifying data structure, my goal was to provide users with systems that can log interaction histories in an interoperable way. The goal was to make interaction histories more easily shareable among the different users' applications. It would also provide users with a more flexible environment that they can customize with their own tools to manipulate interaction histories. The first iteration of the Steps model followed this philosophy and allowed for the implementation of OneTrace, an example of a cross-application interaction history logging system.

But when experimenting with Steps and OneTrace, this first model showed some limitations both in terms of design and clarity to comprehend how to use it and how to make tools that use it. First, it considers relationships between Steps as attributes of these data. As I critiqued in the previous section, relationships are rather constraints applied by instruments on collections of history data to create a new object that makes sense of history. Second, a unifying data structure among interaction history data should make it clear which attributes are editable, and which are only readable. Designers should have a better overview of the accessibility of a Step's attributes in order to design better interaction substrates and instruments out of this model. A new iteration on the first Step model should then contain a set of *core attributes* that any history tool or application can understand and operate on. It should also support personalization by including a set of *context* attributes that users can adapt to their needs.

The core attributes provide a factual description of a user's interaction with the system. An interaction can be described as a user U performing an operation O on the system S at time T. Therefore, the four core attributes describe who (U) did what (O) where (S) when (T). I call them Authorship, Interaction, Source and Time. Core attributes cannot be modified as this would alter the event they represent.

Context attributes, on the other hand, can be altered by users to provide contextual information about the description provided by the core attributes. They should help users make sense of historical data by letting them group traces and create relationships between them. Based on my first study and the two workshops I conducted to design TraceLens and TracePicker, I can already define four requirements that translate into context attributes. First, users should be able to assign some sort of label to traces that represents an abstract task or the user's motivation while leaving a trace. Second, users should be able to express a notion of sequence between these data. Then, users should be able to control the granularity of their history data

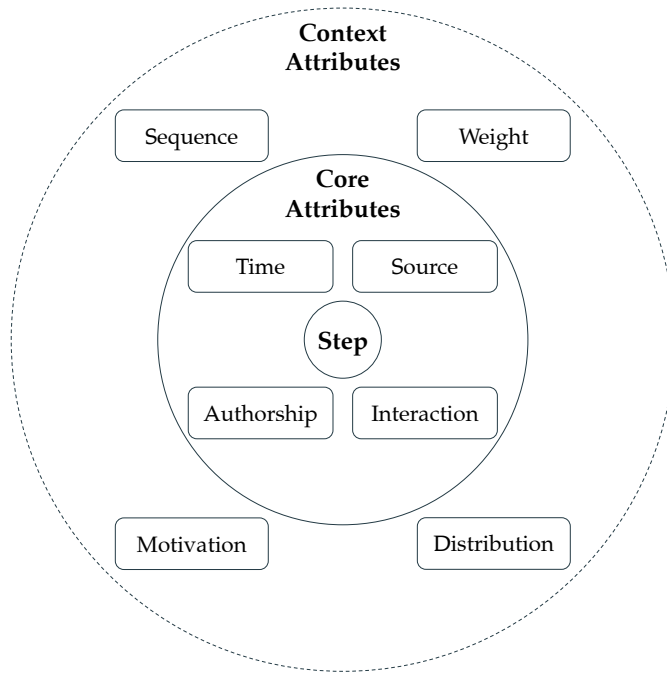


Figure 33: Iteration on the Step unifying data structure composed of two type of attributes : core attributes that can be accessed but not edited, and extensible context attributes that can be edited by users and dedicated tools.

collections and highlight specific traces. Finally, users should be able to provide information about who should have access to a trace, to address privacy issues.

These elements do not conflict with the main philosophy behind Steps. Steps are still self-contained objects that can be used independently of their source application. They can still be manipulated and grouped in a variety of ways, not limited to the traditional timeline or list of commands that are common in current applications. But with this iteration, I want to clarify the extensibility it offers. Indeed, the context attributes that I have identified could be complemented with more specialized attributes depending on use cases. Applications and tools that manipulate Steps should simply leave untouched the attributes they do not recognize. Figure 33 describes this iteration over my former data structure following the aforementioned new requirements. As described before, this new model for Step holds two different types of attributes :

THE CORE ATTRIBUTES: The core attributes of a Step are the attributes that were set at the creation of the history element. They are to be read-only and persistent, as they factually describe the action performed on the system. History tools can read these attributes, but they cannot modify them.

The *"Authorship"* attribute contains the *"who"*, i.e. the user who performed the action.

The *"Interaction"* attribute contains the *"what"*, i.e. the content or description of the command or action that was performed by the user on the system, such as modifying a text or sending an instant message;

The *"Source"* attribute contains the *"where"*, i.e. the application and/or document in which the action was performed, that is the source that logged the action.

The *"Time"* attribute communicates the *"when"*, i.e. the timestamp when the logged action was performed;

THE CONTEXT ATTRIBUTES: The context attributes of a Step are the attributes that can be edited by users to convey additional meaning about the event. These attributes can be set at the creation of a Step, and can be edited by users with dedicated tools.

The *"Motivation"* attribute contains the *"why"* of a Step, i.e. the reason behind the user's action. It can be expressed through different means such as tagging a Step with a color or a label. It provides an additional layer of description at the initiative of the user or a group of collaborators.

The *"Sequence"* attribute contains the information of whether the Step is part of a causal sequence of actions, for example when one event occurred because or in response to another event. This makes it possible to consolidate the context of past actions and to cluster history data into sequences.

The *"Weight"* attribute specifies the importance of a Step. It allows certain history data to stand out, for example by marking one as a higher-level task or as a milestone.

The *"Distribution"* attribute contains the *"whom"*, the users to which the Step should be distributed. It can be a team of individuals, an exhaustive list of persons or a single other user. This one is hard to define for now because it also implies to define to what extent someone having access to a Step might modify the persons allowed to access it. For now, I will just consider the author to be the only one allowed to define to whom the Step is distributed. Depending on roles and work processes, this may need to be refined.

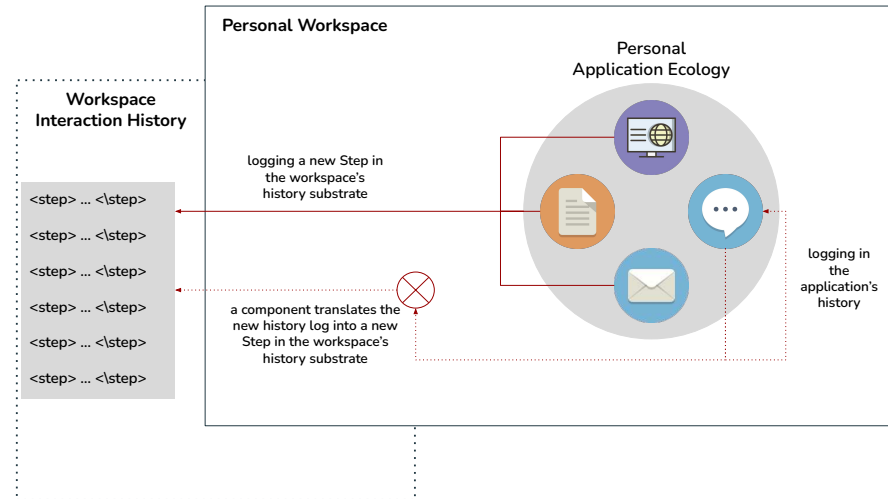


Figure 34: A framework for writing Steps: how the user's workspace, application ecology and their histories interact with each other.

6.3 A GLOBAL FRAMEWORK FOR INTERACTION HISTORY SUBSTRATES AND INSTRUMENTS.

The previous section introduced a new iteration of the Steps unifying data structure. It clarifies how history instruments interact with history substrates based on Steps to create new substrates in the form of collections of interaction history data or other history objects. With this new definition in mind, I want now to define a framework that describes the possible interactions between the workspace, applications and documents, interaction history, history elements, history instruments and history substrates.

Figure 34 shows the first side of this framework which details how Steps are logged. It shows how user's applications from their personal ecology should deal with logging history elements based on the Steps model. The user's workspace is associated with its own interaction history which is a collection of Steps, and thus a first history substrate. Each time the user interacts with one of their applications in the workspace, the application should log history data and thus log a new Step in the workspace's history. This framework is based on the philosophy of malleable software already applied by different works such as Webstrates and Vistrates [6, 43]. It relies on the assumption that a user's workspace is a flexible and extensible environment. The user can integrate any application they may need in their workspace. There are two options for applications to log history data. As a first option, the integrated application knows how to log steps in the workspace's interaction history, and thus follows the previously described behaviour. As a second option, the application has its own way of logging history in its own interaction history, but should provide access to this history. Then, users or the environment

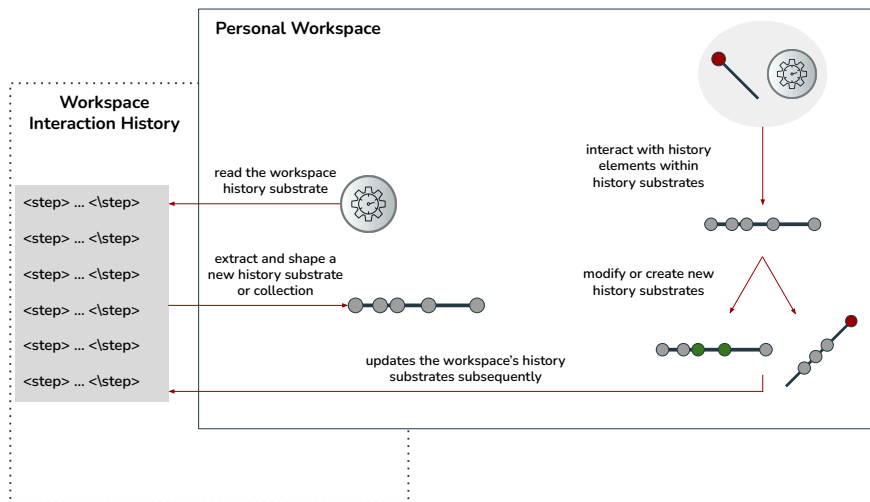


Figure 35: A framework for reading and interacting with Steps: how history instrument, history elements, and history substrates or collections interact with each other.

that the application is included in should be able to add a component that wraps this interaction history log, and translates it into a new history substrate in the workspace's history. In keeping with the philosophy of malleable software, this *"translating"* component may also add a layer of control from the users' side. It may allow them to change some of the information logged in the workspace history, such as its granularity. It should support defining new custom context attributes, such as suggested in the Steps model.

Figure 35 shows the second side of this framework which details how Steps should be read and manipulated by dedicated instrument. It shows how different types of history instruments may interact with history elements and substrates to create or help users create new history substrates. First we have instruments such as a timeline instrument that reads the workspace's history substrate in order to extract and build a new collection of history elements, a new history substrate. This new history substrate is shaped by the relationships applied by the instrument's reified command. Then, we have instruments that can interact with these substrates. They can interact with specific elements in the collection, or with the whole substrate, to modify it or to create a whole new history substrate. These manipulations often translate into changes into the history elements attributes, those defined as *"context attributes"* earlier, and these changes are reflected into the workspace's history substrate. Thanks to unifying structures such as the Steps model presented earlier, instruments manipulating and creating history substrates should provide new collections of history elements with attributes and properties that are understandable by any other instrument designed to manipulate Steps-based history substrates This makes the workspace a powerful poly-

morphic and flexible environment in terms of interactivity. Users may interact with their history with any interaction history tool they like and import them in their workspace without worrying about compatibility issues.

6.4 ITERATING ON THE PROOF-OF-CONCEPT : ONETRACE AND WEBSTRATES

As suggested in the previous section, this work was inspired by the work of Klokrose et al. [43] on Webstrates¹. Webstrates is a platform and a server where the web pages' Document Object Model (DOM) are editable by any client having access to it. Such modifications are synchronized among clients in real time, which enables the creation of shared collaborative web applications. Changes can be applied to a webstrate's DOM, but also to their embedded JavaScript code and CSS styles. All these changes are made persistent on the server and allow for adaptable and customizable web applications.

6.4.1 *Why is Webstrates Relevant?*

The concepts behind Webstrates and some parts of its design make it a perfect choice for implementing a proof-of concept of my proposed framework. Amongst all the features it offers, the main arguments for this choice are the four following ones:

DOM SYNCHRONIZATION. The idea of OneTrace is to show the capabilities of cross-application and cross-user histories. The Webstrates server automatically and instantly synchronizes any changes in the collaborative web applications. If stored within the DOM, Steps can be logged and shared instantly among users working in the same environment. It also allows for a quick development of simple proto-web applications that can be used by multiple users, such as basic WYSIWYG text editors or notepads.

TRANSLUCSION. In Webstrates, to transclude means to load a webstrate into another webstrate using an iframe HTML element. It is interesting for my proof-of-concept because it enables creating an environment where one or multiple users work in a unique context, but want to include and import applications and history tools of their choosing within this workspace. The flexibility of transclusion allows for the personalization of the workspace while providing a unique, shared environment for different users to collaborate. When a webstrate transcludes another webstrate, it also has access to its Document Object Model. Therefore, it can listen to and propagate events to specific nodes of the transcluded page's DOM. This makes it ef-

¹ <https://webstrates.net/>

ficient for the workspace to listen to events triggered when a new Step is created by a transcluded application, or to notify tools and instruments that a new Step has been created.

TRANSIENT DATA. In Webstrates, DOM elements are by default persisted on the server and synchronized across the clients of the same webstrate. This is the opposite of the usual behaviour of regular web applications, which must explicitly persist DOM elements and data, for example before reloading the page. To allow parts of the webstrate's DOM to be temporary and specific to a single client, Webstrates allows to create *transient* elements. Any DOM element added inside a `<transient>` HTML tag becomes a transient element and are not synchronized with the other clients of the page. HTML attributes can also be made transient by prefixing the attribute's name with "transient-". In my case, transient elements help maintain a personal space within a shared work environment. It allows for different users to log in the same shared workspace and open their own applications, tools and instruments in transient windows. It can also help with specific interactions on shared objects. For example, when a user is browsing a timeline, the other users may not want to see the timeline scrolling while the first user tries to find a specific element in the timeline.

SIGNALING. While Webstrates allows DOM elements to be persisted and shared, that is not the case for JavaScript processing and the data that results from it. This means that events cannot be sent between clients of the same Webstrate. To remedy that issue, Webstrates integrates a signaling functionality to send string messages among clients. Clients of the webstrate can signal on and subscribe to DOM elements in order to send and receive messages between clients. This functionality is especially useful because the shared history data of my prototype is stored in the same shared workspace. But if a client generates a step from an application, it will be the only one receiving the event triggered by that application even though the generated step is stored within the shared context. With signals, every client of the context is notified that a new step has been stored or updated by another client, so that they can update other tools if relevant.

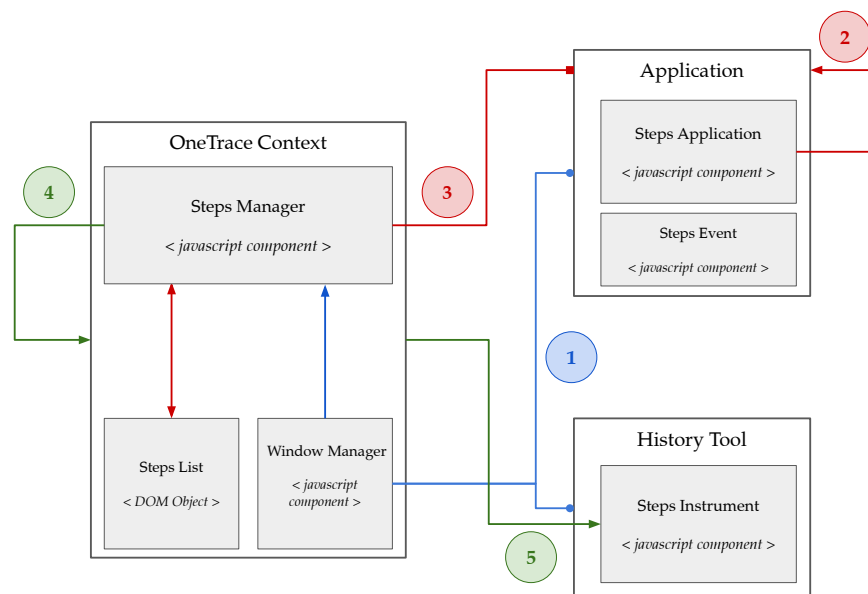


Figure 36: OneTrace implementation with Webstrates : Architecture and Life Cycle. (1) A window manager transcludes applications and history tools in the shared context and connects them to the client's Step Manager; (2) A Steps Application component propagates Steps Event on Applications; (3) The Steps Manager listens to applications' Step Events and updates the shared context Steps List accordingly; (4) The Steps Manager signals all connected clients in the context about the Steps List update; (5) Each Steps Managers propagate a Step Event to the connected history tools.

6.4.2 Implementing OneTrace with Webstrates.

Based on the framework described in the previous section, I now present a new implementation of a OneTrace proof-of-concept history system using Steps as a history data model. Using Webstrates, I implemented the system described in Figure 36. This is a context-based approach, meaning that a context is the workspace in which collaborators work during collaborative activities. In this new system, a context is a webstrate functioning as a shared workspace. It contains three main components : a Steps List stored in the DOM to make it shared and persistent, a Window Manager to load applications and history tools, and a Steps Manager to connect everything together.

The Steps Manager being a JavaScript component, it is unique to each of the context's clients. The Window Manager is mainly using the winbox.js library². It allows the user to open other webstrates in windows within the shared workspace using transclusion. These windows are transient elements and behave as traditional desktop windows. The client's Steps Manager keeps track of the applications

² <https://nextapps-de.github.io/winbox/>

and history tools that are loaded in the workspace by the Window Manager. Note that even though the window DOM element is transient, its content, the loaded application or tool, is a webstrate whose state is synchronized in real-time by the Webstrates server.

An application contains a Steps Application JavaScript component which, when initialized by a webstrate, declares this webstrate as an application in the DOM with the HTML attribute `<step-application>`. The component then propagates Step Events on the application's DOM when a new Step is created. The Steps Manager recognizes an application thanks to the aforementioned HTML attribute. If an application is recognized, it listens to Step Events being propagated on the application's DOM. When a Step Event is received by the Steps Manager, it updates the Steps List in the context.

A history tool contains a Steps Instrument JavaScript component which, when initialized by a webstrate, declares this webstrate as an instrument in the DOM with the HTML attribute `<step-instrument>`. This serves the same purpose as for applications but for the webstrate to be recognized as a history tool by the Steps Manager. The component then listens for Step Events on the history tool's DOM. When it receives a Step Event, it triggers a callback defined by the history tool. Because each client of a workspace contains a single Steps Manager, we use signaling to propagate a Step Event to the history tools so they can update. Every Steps Manager listens to signals on the same DOM element in the shared context. When a Step Manager updates the Steps List in the context, it sends the content of the Step Event as a string message through a signal on the context's DOM. Then, every connected client receives this signal and translates it back to a Step Event. The Step Manager then propagates it to any transcluded webstrate recognized as a history tool.

Structuring the system with JavaScript components combined with the embedded mechanisms of Webstrates makes this new OneTrace implementation very flexible. It better suits the concept of a platform integrating Steps as unified, distributed history data. A collaborative application already developed as a webstrate can easily be turned into a Steps Application by including the corresponding script and instantiating a Step Application object in the main application's script. Then, the *newStep* method of this object can be called to propagate a Step Event to log user events. The Step Event comes with predefined arguments that correspond to the core attributes of the Steps model. However, this also is flexible, as custom events can add new context attributes through inheritance. The Steps Manager receiving the event will dynamically read the event's attribute. The same can be done with history tools or instruments, with the Steps Instrument script. The *stepListener* method of the initialized steps instrument ob-

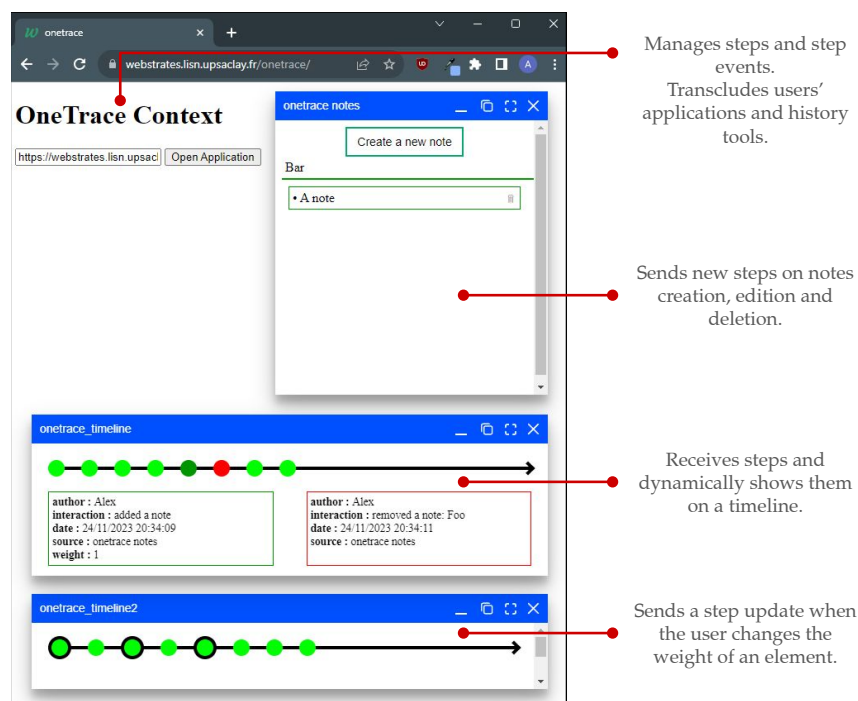


Figure 37: OneTrace implementation with Webstrates : User Interface. Four applications were created as webstrates for this proof-of-concept prototype. The Context application manages the Steps, i.e. a shared workspace in which users can import their own tools and applications. The OneTrace Notes application is a collaborative list of digital post-it notes that generates a Step each time a user modifies, creates or deletes a note. These Steps are gathered by the Context transcluding the Notes application thanks to the Steps Application component. The OneTrace Timeline is a timeline displaying all the Steps from the context transcluding it, which is dynamically updated thanks to the Steps Instrument component. Users can hover elements in this timeline to show details of a Step. A second version of this timeline allows users to interact directly with the Steps and update a context attribute corresponding to their "weight", or what I called granularity in the Steps model. This history tool both includes the Steps Application and the Steps Instrument component. It both updates and is being updated by the Context.

ject needs to be set up with a callback method to react to the new Steps logged in the context.

For both Steps Applications, that log Steps, and Steps Instruments, that read Steps, this structure makes it easy to plug existing webstrates' scripts dealing with history data. It typically only requires a few tweaks to conform to the Steps model thanks to its flexibility. These two scripts are also independent and a webstrates can be declared a "Steps Application" and a "Steps Instrument" for both reading and editing history data, similar to what the TracePicker was designed for.

Figure 37 shows the implementation of this system. The design of the interface and the applications is kept to a minimum as it is only meant to show the system's possibilities. It features a first webstrates called "OneTrace Context"³, which contains the Steps Manager component and holds the Steps List in its DOM. In the user interface, an input field allows the user to open other webstrates in desktop-like windows. It also features a collaborative notepad⁴ allowing users to add and remove digital post-it notes from a shared whiteboard. This collaborative application features the Steps Application component and logs the addition, modifications and removal of notes as Steps. The first history tool⁵ is a timeline that shows the Steps' information when hovered over it in the timeline. The user can also select an element of the timeline for its information to show permanently. This application features the Steps Instrument component to update the timeline when a Step is added or updated in the context. The second history tool⁶ is the same timeline with an additional feature. This timeline allows users to give "*weight*" to a Step by double clicking on it. When an element in this timeline is "*heavier*", it shows by being magnified. This application features both the Steps Application component to update the Steps' weight in the context, and the Steps Instrument component. Whenever a Step is logged or updated in the context via the Steps Applications, it is propagated accordingly to all Steps Instruments. The applications mentioned in this section and their source code can be found following the links in the footnotes.

6.5 CONCLUSION AND DISCUSSION

In this chapter, I revisited my first observations and experiments to question what really is history as a first class object that users can interact with. It led me to better define the differences between interaction history, history instruments and the history collections resulting from the interaction between the two. The goal of this reflection is to attempt to define what makes interaction history a material that users can interact with and shape as needed, in other words what makes it an interaction substrate. It led me to define interaction history as a raw collection of history elements which I describe as Steps. Users extract Steps and put it together as new collections of history elements, in a way that makes sense to them, depending on the context. To achieve this, they use history instruments that will make visible or create new relationships among relevant history elements. I state that these collections of Steps are history substrates, a container of history elements that reifies relationships between Steps.

³ <https://webstrates.lisn.upsaclay.fr/onetrace/>

⁴ https://webstrates.lisn.upsaclay.fr/onetrace_notes/

⁵ https://webstrates.lisn.upsaclay.fr/onetrace_timeline/

⁶ https://webstrates.lisn.upsaclay.fr/onetrace_timeline2/

As a result, I iterated on my Steps unifying model for history data. Due to this new way of defining interaction between history substrates and instruments, relationships could not be a part of the structure encapsulating history data attributes. The new model follows this philosophy of both supporting the integrity that history data needs through its core attributes, and providing the flexibility users need to fit their history usage depending on their needs through extensible context attributes. Steps attributes are now a way for instruments to build new collections, or new history substrates, by drawing relationships among Steps.

I then presented a framework that describes an environment based on history substrates and instruments. It describes how interaction history, history tools, history collections and context of use all interact together in this environment. In line with the interaction substrates theory and related works on malleable software, it aims at providing a sound basis for designing a system that supports users in reclaiming control over their work environment. The first iteration of OneTrace implemented parts of this framework, focusing on a unified way to log history data no matter the application logging it. But when it came to instruments, and especially the history substrates resulting from their use, I did not design the polymorphic capabilities that I describe in this revised framework.

As a proof-of-concept system for this framework, I implemented a new version of OneTrace supported by the Webstrates environment. It allowed me to implement the flexible environment I described, in which users can collaborate in the same workspace synchronously. In this workspace, they can also import their own tools and applications with respect to their personal application ecologies, and still be able to share their traces as Steps. This system provides tools and components to implement more advanced interaction history systems shared amongst users and applications as an independent digital material. I believe this goes further than earlier related work that addressed the goals of enabling flexible cross-application interaction histories. For example, although Rekimoto's Time Machine Computing [69] shows the benefits of unifying histories between applications in a single system, it also offers no common structure for histories to design new instruments, and does not take into account multi-user working environments. The Leavo system by Jeuris et al. [37] is also an important contribution, but it does not take into account the flexibility of the user's personal space and offers only one timeline for interacting with one's history rather than providing users with different ways of manipulating their traces. With the notion of history substrates that I described, interaction history is now an independent, interactive, polymorphic and shareable object. I hope this

will open discussions about how to design better interaction history instruments but also more flexible interactive systems in general.

Still, some limitations remain with this approach. By centering the history framework around workspaces, I limited the scope of interaction between all its elements to a single joint activity. I took this direction in order to explore the feasibility of such a system. However, it needs to be iterated on using knowledge from related work about flexible and distributed workspaces, such as the work I described about collaborative writing applications and traces ecology. The scope must be extended to tasks and activities beyond the main object of the collaboration. For example, further observations should be made on how people want to reuse histories from one activity to another, or how users including external content with its own history in a shared workspace want to deal with the potential conflicts. It also requires further analysis of how histories from different activities can be linked together, and how users can control how to share their histories with other team members.

Nevertheless, this new system offers a fully collaborative environment which is robust enough, and flexible enough to be evaluated in various use cases and with less rigid technical requirements. In future work, we may conduct longer term evaluations by providing a workspace for participants to log into anywhere, and develop applications and instruments for OneTrace depending on the use case we want to study. Further evaluations of the system and of cross-application history's impact on work habits will then show less limitations than the previous studies I conducted with TraceLens and TracePicker.

CONCLUSION

7.1 OVERVIEW OF THE CONTRIBUTIONS

In this thesis, I explored how to improve current interaction history tools in order to better support users' interactions with their traces. I reviewed previous work about interaction histories, and especially those that aimed at better understanding how users actually interact with traces beyond only browsing them. It helped me define interaction history not only as raw data, but also as digital objects with their own affordances that can be manipulated and shaped by the users to some extent. I learned that people seek to enrich their past history with information that is not tied to their document of interest or the applications they use and provide a better understanding of the context in which traces were left. I also learned that people often need to cross-reference histories from different systems to do so, for example, for re-documenting a specific activity that occurred among different applications in their workspace. In that case, current interaction history tools prevent historical data from being used by other systems, thereby hindering users' activities when it comes to interacting with traces. These points led me to state that cross-application interaction histories should be paid more attention to when it comes to designing interaction histories.

Reviewing the literature on coordination in collaborative writing settings, I showed that interaction histories play a huge role in communicating and understanding a team's progress on common activities. However, current interactive systems barely support sharing interaction histories among team members' application ecologies and people have to come up with their own workarounds to support collaborative practices that are not supported by their current tools. To illustrate and better understand this environment of traces confined in their source applications, I described the collaborative writing traces ecologies framework which draws inspiration from previous work on artifact ecologies [48, 61]. Based on this framework and the reviewed literature, I conducted story interviews with 12 knowledge workers about their history use in collaborative writing settings in order ground this knowledge in a specific use case. I wanted to better understand how people use histories for coordination, and how their current tools fail to support their activities. I analyzed the results of these interviews by performing a thematic analysis with the help of another researcher. The results showed that participants mainly interact with their traces in two main ways. First, as independent objects,

they customize traces to differentiate them from one another, in order to emphasize the meaning behind a trace, or to add contextual cues to better understand it. Then, as collections of history elements, participants create new artifacts to re-document past activities, to make it more understandable for them and their team. This requires browsing across different histories, and merging and extracting relevant information to compile it as new content. Participants also expressed numerous issues with their tools that hinder these practices. Participants have to browse histories in numerous places and sometimes in huge collections, which requires more work to identify relevant history data. They also often expressed having no means to consolidate the meaning of a past event, to understand why an action occurred on top of the pragmatic details of these actions.

The empirical findings from this first study showed that while users need cross-application history tools to support their collaborative practices, they are widely unsupported by current tools. These results concurred with my first assumptions based on the literature that interaction histories were not flexible enough to support users' needs. To answer this lack of flexibility and allow the design of a cross-application history system, I introduced a new approach for designing interaction history systems, tools, and objects. With the knowledge I previously gathered, I introduced the Step structure, a unifying history data structure. I then introduced OneTrace, an example of how to implement a cross-application interaction history system using the Step unifying history data structure. OneTrace is both a universal logging system using Steps, and a user interface that implements basic applications such as a text editor, a messaging application and a mail client. It aims to implement various history instruments that allow cross-application interaction history manipulation and to experiment with them.

In order to show the capabilities of OneTrace's cross-application histories to support users with a new kind of history tools, I conducted two projects in parallel based on the data extracted from the first study. I stated two different design goals and conducted various participatory design activities with users to answer these goals, including brainstorming and prototyping. I also used generative design tools such as the instrumental interaction framework. These activities led to the design of TraceLens, an instrument to improve cross-application history browsing, and TracePicker, an instrument to help users customize collections of history elements and enhance past actions understanding. I implemented both designs in the OneTrace proof-of-concept system as examples of cross-application history tools. I assessed these instruments with observation- and scenario-based user studies to evaluate if each of the tools fulfilled their goals, based on the participants' perceived improvements when com-

paring it to their usual system. Results showed that a unified history can indeed improve awareness of the timeline in a collaborative project and help users appropriate history data and generate new meaningful artifacts that enhance coordination.

OneTrace and Steps allowed me to design cross-application history instruments and showed that this kind of system allows for a more flexible and better suited environment when it comes to manipulating interaction histories. But experimenting with it made me reflect on the lack of consistency of the solutions I provided. As a result, I formulated several critiques that allowed me to iterate on the Steps data structure and make it more universal. I also needed to provide a framework defining how the different digital objects in this kind of environment should interact with each other. To this end, I turned towards previous work that describes and implements flexible software and interactive systems, and especially the substrates and instruments interaction model described by Beaudouin-Lafon [10, 11]. It helped me describe a framework consisting of a set of guidelines in an environment composed of history data and graphical substrates, history substrates collections, and history instruments. The aim of this framework is to provide a better understanding of how to support cross-application histories in future systems by describing how each element of this environment may interact with each other. It should help design graphical substrates that reify history data into customizable objects, and relevant history data attributes into affordances. It also shows how to design polymorphic history instruments that take full advantage of these affordances to support users in creating new substrates as collections of history elements that reify meaningful relationships among history elements. To demonstrate the potential of such a system based on this new framework, I iterated on my proof-of-concept and implemented a new version of OneTrace using the Webstrates technology.

The work I conducted during this thesis provides a better understanding of how current interactive systems hinder interaction history use. While most related work on improving interaction histories focus on single applications, or on web browsing due to the lighter needs for implementing new tools, the framework I introduced opens the way for thinking of interaction histories under a new point of view. The new OneTrace implementation based on Webstrates also shows that implementing such systems could be quite simple thanks to state-of-the-art technologies. It complements related work aiming at implementing new interactive systems that free themselves from the dominant WIMP interaction model. It describes a way to provide users with interaction histories as objects they can manipulate rather than interaction histories as data they have little to no control over.

7.2 LIMITATIONS AND FUTURE WORK

The first limitation of my work is related to the use case I investigated. I only studied collaborative document editing and based the next steps of my works on this use case. More use cases of collaboration around common objects such as videos, slideshows or even graphical objects are to be investigated to gather more observations about history use. I also emphasized collaborative settings because it included both personal and shared use of histories, but observing a single user environment is also an option for future work, to see if different practices around traces emerge. The benefits of investigating a larger set of use cases would be to mainly challenge my Steps unifying structure for history data, especially when it comes to the malleable attributes that I described. These attributes are the transcription of users' needs when it comes to regrouping or customizing history data, and in order to generalize such a data structure, it would be necessary to verify their validity in various use cases and see if we can extend the list of these attributes.

When it comes to the evaluation of the two instruments I designed for OneTrace, although the aim was not to provide users with a working environment, it showed some limitations that I discussed in the dedicated chapter. Especially due to the technical limitations of OneTrace, I could not provide users with an environment stable enough to be used in the wild outside of a controlled environment. Interaction history being a support of users' memory, it would be beneficial to run longer term field studies to assess the impact of cross-application histories and dedicated history instruments on users information retrieval. Having a stable system would also allow for a more reliable comparison with a control group of participants using their everyday tools, which was not possible with my current implementation.

About my final descriptive framework, two aspects still remain to be studied further. First, I took into account the distributed aspect of interaction histories in collaborative settings, but I did not provide guidelines on how users should be able to actively choose how to share history data when it is relevant, or how designers should take into account and support the notion of roles in collaborative settings. The main reason for this is that I lacked information on this two points to provide well-grounded guidelines. Because I think that users should remain in complete control over the data they share, future work should investigate history data sharing and define the associated guidelines. The second aspect I would like to investigate further is the notion of work contexts. Indeed, my current framework associates interaction history data substrates with a workspace. But related work on tasks and activities analysis showed that the same workspace can be the theater of several activities in parallel. Thus, fu-

ture work should study how interaction history should be logged in such work environments, and whether context-centric histories would be more beneficial than workspace-centric histories. It should also shed light on how and why interaction history data should be distributed among contexts.

On a more technical level, I would like to build on the Webstrates version of my proof-of-concept prototype, to make it more robust and to develop new tools with it. For example, it is possible to implement a version of TraceLens and TracePicker with it, and it would be really useful to investigate further and replicate the studies about these two tools in a new context. I would also like to investigate the issues encountered when trying to adapt existing tools to this system. Indeed, the philosophy behind the Webstrates system is to provide a toolkit to implement distributed and synchronized systems, and the benefits of such systems is the distributed and malleable aspect of it. I tried to stick to this original idea while also trying to fit my own research on how interaction history should be designed, and I would like to study my system's adaptability with existing systems.

Finally, I would like to see to what extent my concept of Steps as a universal description of history data could be extended to any type of temporal data. As an example, I participated in the development of a video prototyping tool called Collaborative VideoClipper [53] based on Webstrates. This application lets users upload video clips and organize them on timelines as "*storylines*". They can also add text-based elements on these timelines as "*titlecards*" to add a layer of description to the video clips. Then, the application generates a full video prototype by compiling all the elements on the timeline. I would like to try and implement it with OneTrace to investigate the differences between temporal data from a video and those from histories, since they can both be displayed on timelines. I think it would help reflect on my Steps description of historical data and maybe extend it as a universal description of temporal data.

The initial observation that led to my research is one that could apply to many of interactive systems. By design, the data generated by the use of applications and tools are often intended to be only used by the source application that generated them. Users rarely have the power to customize their workspace with the tools of their choosing, resulting in having too many tools imposed on them. While specific software suites such as Microsoft Office or Google Workspace tend to create bridges among their own applications, it remains confined into these software suites. The issue is the same with interaction histories, and the impact is arguably stronger on user's work processes. The efficient use of historical data lies in the capabilities of cross-referencing histories. In the case of collaborative activities, a facilitated use of histories makes coordination strategies more efficient. But in the current

systems, despite their factual necessity, interaction histories are at the very best a basic facility destined to only show a fraction of the information needed to understand the life-cycle of the system. It makes interaction histories underused because it does not fulfill users' needs. In the future, to make better interactive systems, interaction histories should be designed by studying end users' practices, and not just be seen as an informative list of items. This would be a first step into making interaction history tools more independent, flexible and most of all relevant. It also means that we should provide users with complete control over their history data. I believe that the future of interaction histories lies in the same place as interactive systems in general, with flexible and extensible interactive environments which users can appropriate and control with the tools of their choosing.

BIBLIOGRAPHY

- [1] Cambridge Dictionary. <https://dictionary.cambridge.org/>. Last visited 16 March 2023.
- [2] Oxford Learner’s Dictionaries. <https://www.oxfordlearnersdictionaries.com/>. Last visited 16 March 2023.
- [3] Muhammad Adnan, Mike Just, and Lynne Baillie. “Investigating Time Series Visualisations to Improve the User Experience.” In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. CHI ’16. San Jose, California, USA: Association for Computing Machinery, 2016, 5444–5455. ISBN: 9781450333627. DOI: [10.1145/2858036.2858300](https://doi.org/10.1145/2858036.2858300). URL: <https://doi.org/10.1145/2858036.2858300>.
- [4] Jason Alexander, Andy Cockburn, Stephen Fitchett, Carl Gutwin, and Saul Greenberg. “Revisiting read wear: analysis, design, and evaluation of a footprints scrollbar.” In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2009, pp. 1665–1674. DOI: [10.1145/1518701.1518957](https://doi.org/10.1145/1518701.1518957).
- [5] Mirzel Avdic, Susanne Bødker, and Ida Larsen-Ledet. “Two Cases for Traces: A Theoretical Framing of Mediated Joint Activity.” In: *Proceedings of the ACM on Human-Computer Interaction* 5.CSCW1 (2021), pp. 1–28. DOI: [10.1145/3449289](https://doi.org/10.1145/3449289).
- [6] Sriram Karthik Badam, Andreas Mathisen, Roman Rädle, Clemens N. Klokmoose, and Niklas Elmqvist. “Vistrates: A Component Model for Ubiquitous Analytics.” In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (2019), pp. 586–596. DOI: [10.1109/TVCG.2018.2865144](https://doi.org/10.1109/TVCG.2018.2865144).
- [7] Cornelis B Bakker and Marianne K Bakker-Rabdau. *No trespassing!: Explorations in human territoriality*. San Francisco: Chandler & Sharp Publishers, 1973.
- [8] Alexandre Battut, Kevin Ratovo, and Michel Beaudouin-Lafon. “OneTrace: Improving Event Recall and Coordination With Cross-Application Interaction Histories.” In: *International Journal of Human-Computer Interaction* (2024), pp. 1–18. DOI: [10.1080/10447318.2024.2332848](https://doi.org/10.1080/10447318.2024.2332848).
- [9] Michel Beaudouin-Lafon. “Instrumental interaction: an interaction model for designing post-WIMP user interfaces.” In: *Proceedings of the SIGCHI Conference on Human factors in computing systems*. New York, NY, USA: ACM, 2000, pp. 446–453. DOI: [10.1145/332040.332473](https://doi.org/10.1145/332040.332473).

- [10] Michel Beaudouin-Lafon. "Towards unified principles of interaction." In: *Proceedings of the 12th Biannual Conference on Italian SIGCHI Chapter*. 2017, pp. 1–2. DOI: [10.1145/3125571.3125602](https://doi.org/10.1145/3125571.3125602).
- [11] Michel Beaudouin-Lafon. "Au-delà des applications : Substrats et instruments d'interaction." IHM 2023 - 34e Conférence Internationale Francophone sur l'Interaction Humain-MachineAFIHM, Université de Technologie de Troyes. Mar. 2023. URL: <https://hal.science/hal-04014963>.
- [12] Michel Beaudouin-Lafon, Susanne Bødker, and Wendy E Mackay. "Generative theories of interaction." In: *ACM Transactions on Computer-Human Interaction (TOCHI)* 28.6 (2021), pp. 1–54. DOI: [10.1145/3468505](https://doi.org/10.1145/3468505).
- [13] Michel Beaudouin-Lafon and Wendy E Mackay. "Reification, polymorphism and reuse: three principles for designing visual interfaces." In: *Proceedings of the working conference on Advanced visual interfaces*. New York, NY, USA: ACM, 2000, pp. 102–109. DOI: [10.1145/345513.345267](https://doi.org/10.1145/345513.345267).
- [14] Amaury Belin and Yannick Prié. "DIAM: Towards a Model for Describing Appropriation Processes through the Evolution of Digital Artifacts." In: *Proceedings of the Designing Interactive Systems Conference*. DIS '12. Newcastle Upon Tyne, United Kingdom: ACM, 2012, 645–654. ISBN: 9781450312103. DOI: [10.1145/2317956.2318053](https://doi.org/10.1145/2317956.2318053). URL: <https://doi-org.ins2i.bib.cnrs.fr/10.1145/2317956.2318053>.
- [15] Ofer Bergman, Ruth Beyth-Marom, and Rafi Nachmias. "The Project Fragmentation Problem in Personal Information Management." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '06. Montréal, Québec, Canada: Association for Computing Machinery, 2006, 271–274. ISBN: 1595933727. DOI: [10.1145/1124772.1124813](https://doi.org/10.1145/1124772.1124813). URL: <https://doi.org/10.1145/1124772.1124813>.
- [16] Susanne Bødker and Clemens Nylandsted Klokmoose. "Dynamics in artifact ecologies." In: *Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design*. New York, NY, USA: ACM, 2012, pp. 448–457. DOI: [10.1145/2399016.2399085](https://doi.org/10.1145/2399016.2399085).
- [17] Marcel Borowski, Luke Murray, Rolf Bagge, Janus Bager Kristensen, Arvind Satyanarayan, and Clemens N. Klokmoose. "Varv: Reprogrammable Interactive Software as a Declarative Data Structure." In: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. CHI '22. 2022. DOI: [10.1145/3491102.3502064](https://doi.org/10.1145/3491102.3502064).

- [18] Marcel Borowski, Roman Rädle, and Clemens N. Klokmoose. "Codestrate Packages: An Alternative to "One-Size-Fits-All" Software." In: *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI EA '18. 2018. DOI: [10.1145/3170427.3188563](https://doi.org/10.1145/3170427.3188563).
- [19] Virginia Braun and Victoria Clarke. "Reflecting on reflexive thematic analysis." In: *Qualitative research in sport, exercise and health* 11.4 (2019), pp. 589–597. DOI: [10.1080/2159676X.2019.1628806](https://doi.org/10.1080/2159676X.2019.1628806).
- [20] John Brown and Paul Duguid. "Organizational Learning and Communities of Practice: Toward a Unified View of Working, Learning, and Innovation." In: *Organization science* 2 (Feb. 1991), 40–57–40–57. DOI: [10.1287/orsc.2.1.40](https://doi.org/10.1287/orsc.2.1.40).
- [21] Nela Brown and Tony Stockman. "Examining the Use of Thematic Analysis as a Tool for Informing Design of New Family Communication Technologies." In: *Proceedings of the 27th International BCS Human Computer Interaction Conference*. BCS-HCI '13. London, UK: BCS, 2013. DOI: [10.14236/ewic/HCI2013.30](https://doi.org/10.14236/ewic/HCI2013.30).
- [22] Rubiela Carrillo, Elise Lavoué, and Yannick Prié. "Towards Qualitative Insights for Visualizing Student Engagement in Web-Based Learning Environments." In: *Proceedings of the 25th International Conference Companion on World Wide Web*. WWW '16 Companion. Montréal, Québec, Canada: International World Wide Web Conferences Steering Committee, 2016, 893–898. ISBN: 9781450341448. DOI: [10.1145/2872518.2891076](https://doi.org/10.1145/2872518.2891076). URL: <https://doi-org.ins2i.bib.cnrs.fr/10.1145/2872518.2891076>.
- [23] Marianela Ciolfi Felice, Nolwenn Maudet, Wendy E. Mackay, and Michel Beaudouin-Lafon. "Beyond Snapping: Persistent, Tweakable Alignment and Distribution with StickyLines." In: *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. UIST '16. Tokyo, Japan: Association for Computing Machinery, 2016, 133–144. ISBN: 9781450341899. DOI: [10.1145/2984511.2984577](https://doi.org/10.1145/2984511.2984577). URL: <https://doi.org/10.1145/2984511.2984577>.
- [24] Alan John Dix. "Designing for appropriation." In: *British Computer Society Conference on Human-Computer Interaction*. 2007.
- [25] Eric Freeman and David Gelernter. "Lifestreams: A storage model for personal data." In: *ACM SIGMOD Record* 25.1 (1996), pp. 80–86. DOI: [10.1145/381854.381893](https://doi.org/10.1145/381854.381893).
- [26] Jérémie Garcia, Theophanis Tsandilas, Carlos Agon, and Wendy Mackay. "Interactive Paper Substrates to Support Musical Creation." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '12. Austin, Texas, USA: Association for Computing Machinery, 2012, 1825–1828. ISBN: 9781450310154.

- DOI: [10.1145/2207676.2208316](https://doi.org/10.1145/2207676.2208316). URL: <https://doi.org/10.1145/2207676.2208316>.
- [27] Eleanor Jack Gibson. "Principles of perceptual learning and development." In: (1969).
- [28] J.J. Gibson. *The Ecological Approach to Visual Perception*. Resources for ecological psychology. Lawrence Erlbaum Associates, 1986. ISBN: 9780898599596. URL: <https://books.google.fr/books?id=DrhCCWmJpWUC>.
- [29] Tovi Grossman, Justin Matejka, and George Fitzmaurice. "Chronicle: capture, exploration, and playback of document workflow histories." In: *Proceedings of the 23rd annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA: ACM, 2010, pp. 143–152. DOI: [10.1145/1866029.1866054](https://doi.org/10.1145/1866029.1866054).
- [30] Viktor Gustafsson, Benjamin Holme, and Wendy E. Mackay. "Narrative Substrates: Reifying and Managing Emergent Narratives in Persistent Game Worlds." In: *Proceedings of the 15th International Conference on the Foundations of Digital Games*. FDG '20. Bugibba, Malta: Association for Computing Machinery, 2020. ISBN: 9781450388078. DOI: [10.1145/3402942.3403015](https://doi.org/10.1145/3402942.3403015). URL: <https://doi.org/10.1145/3402942.3403015>.
- [31] Joshua Hailpern, Nicholas Jitkoff, Andrew Warr, Karrie Karahalios, Robert Sesek, and Nik Shkrob. "YouPivot: improving recall with contextual search." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2011, pp. 1521–1530. DOI: [10.1145/1978942.1979165](https://doi.org/10.1145/1978942.1979165).
- [32] Han L. Han, Miguel A. Renom, Wendy E. Mackay, and Michel Beaudouin-Lafon. "Textlets: Supporting Constraints and Consistency in Text Documents." In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI '20. Honolulu, HI, USA: Association for Computing Machinery, 2020, 1–13. ISBN: 9781450367080. DOI: [10.1145/3313831.3376804](https://doi.org/10.1145/3313831.3376804). URL: <https://doi.org/10.1145/3313831.3376804>.
- [33] Han L. Han, Junhang Yu, Raphael Bournet, Alexandre Ciorascu, Wendy E. Mackay, and Michel Beaudouin-Lafon. "Passages: Interacting with Text Across Documents." In: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. CHI '22. New Orleans, LA, USA: Association for Computing Machinery, 2022. ISBN: 9781450391573. DOI: [10.1145/3491102.3502052](https://doi.org/10.1145/3491102.3502052). URL: <https://doi.org/10.1145/3491102.3502052>.
- [34] Robert XD Hawkins, Megumi Sano, Noah D Goodman, and Judith W Fan. "Disentangling contributions of visual information and interaction history in the formation of graphical conventions." In: *Cognitive Science Society*. 2019, pp. 415–421.

- [35] William C Hill, James D Hollan, Dave Wroblewski, and Tim McCandless. "Edit wear and read wear." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 1992, pp. 3–9. DOI: [10 . 1145 / 142750 . 142751](https://doi.org/10.1145/142750.142751).
- [36] Donghan Hu and Sang Won Lee. "ScreenTrack: Using a Visual History of a Computer Screen to Retrieve Documents and Web Pages." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2020, pp. 1–13. DOI: [10 . 1145 / 3313831 . 3376753](https://doi.org/10.1145/3313831.3376753).
- [37] Steven Jeuris, Steven Houben, and Jakob Bardram. "Laevo: A Temporal Desktop Interface for Integrated Knowledge Work." In: *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*. UIST '14. Honolulu, Hawaii, USA: Association for Computing Machinery, 2014, 679–688. ISBN: 9781450330695. DOI: [10 . 1145 / 2642918 . 2647391](https://doi.org/10.1145/2642918.2647391). URL: [https://doi.org/10 . 1145/2642918.2647391](https://doi.org/10.1145/2642918.2647391).
- [38] Heekyoung Jung, Erik Stolterman, Will Ryan, Tonya Thompson, and Marty Siegel. "Toward a Framework for Ecologies of Artifacts: How Are Digital Artifacts Interconnected within a Personal Life?" In: *Proceedings of the 5th Nordic Conference on Human-Computer Interaction: Building Bridges*. NordiCHI '08. Lund, Sweden: Association for Computing Machinery, 2008, 201–210. ISBN: 9781595937049. DOI: [10 . 1145 / 1463160 . 1463182](https://doi.org/10.1145/1463160.1463182). URL: <https://doi.org/10.1145/1463160.1463182>.
- [39] Victor Kaptelinin. "UMEA: Translating Interaction Histories into Project Contexts." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '03. Ft. Lauderdale, Florida, USA: Association for Computing Machinery, 2003, 353–360. ISBN: 1581136307. DOI: [10 . 1145 / 642611 . 642673](https://doi.org/10.1145/642611.642673). URL: <https://doi.org/10.1145/642611.642673>.
- [40] A. Kay and A. Goldberg. "Personal Dynamic Media." In: *Computer* 10.3 (1977), pp. 31–41. DOI: [10 . 1109 / C - M . 1977 . 217672](https://doi.org/10.1109/C-M.1977.217672).
- [41] Mary Beth Kery, Amber Horvath, and Brad A Myers. "Variolite: Supporting Exploratory Programming by Data Scientists." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Vol. 10. New York, NY, USA: ACM, 2017, pp. 3025453–3025626. DOI: [10 . 1145 / 3025453 . 3025626](https://doi.org/10.1145/3025453.3025626).
- [42] Mary Beth Kery, Bonnie E John, Patrick O'Flaherty, Amber Horvath, and Brad A Myers. "Towards effective foraging by data scientists to find past analysis choices." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2019, pp. 1–13. DOI: [10 . 1145 / 3290605 . 3300322](https://doi.org/10.1145/3290605.3300322).

- [43] Clemens N Klokmoose, James R Eagan, Siemen Baader, Wendy Mackay, and Michel Beaudouin-Lafon. "Webstrates: shareable dynamic media." In: *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA: ACM, 2015, pp. 280–290. DOI: [10.1145/2807442.2807446](https://doi.org/10.1145/2807442.2807446).
- [44] Clemens N Klokmoose, Christian Remy, Janus Bager Kristensen, Rolf Bagge, Michel Beaudouin-Lafon, and Wendy Mackay. "Videostrates: Collaborative, Distributed and Programmable Video Manipulation." In: *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA: ACM, 2019, pp. 233–247. DOI: [10.1145/3332165.3347912](https://doi.org/10.1145/3332165.3347912).
- [45] Ida Larsen-Ledet. "Understanding and Supporting Collaborative Academic Writing as a Fragmented Process." In: *Proceedings of 17th European Conference on Computer-Supported Cooperative Work-Doctoral Colloquium*. European Society for Socially Embedded Technologies (EUSSET). 2019. DOI: [10.18420/ecscw2019_dc6](https://doi.org/10.18420/ecscw2019_dc6).
- [46] Ida Larsen-Ledet and Marcel Borowski. "'It Looks Like You Don't Agree': Idiosyncratic Practices and Preferences in Collaborative Writing." In: Dec. 2020, pp. 339–354. DOI: [10.1145/3441000.3441032](https://doi.org/10.1145/3441000.3441032).
- [47] Ida Larsen-Ledet and Henrik Korsgaard. "Territorial Functioning in Collaborative Writing: Fragmented Exchanges and Common Outcomes." In: *Computer Supported Cooperative Work (CSCW)* 28 (May 2019). DOI: [10.1007/s10606-019-09359-8](https://doi.org/10.1007/s10606-019-09359-8).
- [48] Ida Larsen-Ledet, Henrik Korsgaard, and Susanne Bødker. "Collaborative writing across multiple artifact ecologies." In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2020, pp. 1–14. DOI: [10.1145/3313831.3376422](https://doi.org/10.1145/3313831.3376422).
- [49] Siân E Lindley, Gavin Smyth, Robert Corish, Anastasia Loukianov, Michael Golembewski, Ewa A Luger, and Abigail Sellen. "Exploring new metaphors for a networked world through the file biography." In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2018, pp. 1–12. DOI: [10.1145/3173574.3173692](https://doi.org/10.1145/3173574.3173692).
- [50] Robert Lokaiczuk and Manuel Goertz. "Extending low level context events by data aggregation." In: *Proceedings of I-KNOW*. Vol. 8. 2008, pp. 118–125.
- [51] Walid Maalej and Alexander Sahm. "Assisting engineers in switching artifacts by using task semantic and interaction history." In: *Proceedings of the 2nd International Workshop on Recommendation Systems for Software Engineering*. New York, NY, USA: ACM, 2010, pp. 59–63. DOI: [10.1145/1808920.1808935](https://doi.org/10.1145/1808920.1808935).

- [52] Wendy E Mackay. "Designing with sticky notes." In: *Sticky Creativity*. Elsevier, 2020, pp. 231–256. DOI: [10.1016/B978-0-12-816566-9.00010-0](https://doi.org/10.1016/B978-0-12-816566-9.00010-0).
- [53] Wendy E. Mackay, Alexandre Battut, Germàn Leiva, and Michel Beaudouin-Lafon. "VideoClipper: Rapid Prototyping with the "Editing-in-the-Camera" Method." In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. CHI '24. Honolulu, HI, USA: Association for Computing Machinery, 2024. ISBN: 9798400703300. DOI: [10.1145/3613904.3642458](https://doi.org/10.1145/3613904.3642458). URL: <https://doi.org/10.1145/3613904.3642458>.
- [54] Wendy E. Mackay and Anne-Laure Fayard. "HCI, Natural Science and Design: A Framework for Triangulation across Disciplines." In: *Proceedings of the 2nd Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques*. DIS '97. Amsterdam, The Netherlands: Association for Computing Machinery, 1997, 223–234. ISBN: 0897918630. DOI: [10.1145/263552.263612](https://doi.org/10.1145/263552.263612). URL: <https://doi.org/10.1145/263552.263612>.
- [55] Thomas W. Malone. "How Do People Organize Their Desks? Implications for the Design of Office Information Systems." In: *ACM Trans. Inf. Syst.* 1.1 (1983), 99–112. ISSN: 1046-8188. DOI: [10.1145/357423.357430](https://doi.org/10.1145/357423.357430). URL: <https://doi.org/10.1145/357423.357430>.
- [56] Thomas W Malone and Kevin Crowston. "The interdisciplinary study of coordination." In: *ACM Computing Surveys (CSUR)* 26.1 (1994), pp. 87–119. DOI: [10.1145/174666.174668](https://doi.org/10.1145/174666.174668).
- [57] R. Morteo, V.M. Gonzalez, J. Favela, and G. Mark. "Sphere Juggler: fast context retrieval in support of working spheres." In: *Proceedings of the Fifth Mexican International Conference in Computer Science, 2004. ENC 2004*. 2004, pp. 361–367. DOI: [10.1109/ENC.2004.1342628](https://doi.org/10.1109/ENC.2004.1342628).
- [58] Ahmed E. Mostafa, Ehud Sharlin, and Mario Costa Sousa. "ReflectiveHUD: Designing Spatial Interaction History." In: *Proceedings of the 2017 ACM Conference Companion Publication on Designing Interactive Systems*. DIS '17 Companion. Edinburgh, United Kingdom: Association for Computing Machinery, 2017, 233–237. ISBN: 9781450349918. DOI: [10.1145/3064857.3079152](https://doi.org/10.1145/3064857.3079152). URL: <https://doi.org/10.1145/3064857.3079152>.
- [59] Mathieu Nancel and Andy Cockburn. "Causality: A conceptual model of interaction history." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2014, pp. 1777–1786. DOI: [10.1145/2556288.2556990](https://doi.org/10.1145/2556288.2556990).

- [60] Phong H Nguyen, Kai Xu, Andy Bardill, Betul Salman, Kate Herd, and BL William Wong. "SenseMap: Supporting browser-based online sensemaking through analytic provenance." In: *2016 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, 2016, pp. 91–100. DOI: [10.1109/VAST.2016.7883515](https://doi.org/10.1109/VAST.2016.7883515).
- [61] Midas Nouwens, Carla F. Griggio, and Wendy E. Mackay. "'WhatsApp is for Family; Messenger is for Friends': Communication Places in App Ecosystems." In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI '17. Denver, Colorado, USA: Association for Computing Machinery, 2017, 727–735. ISBN: 9781450346559. DOI: [10.1145/3025453.3025484](https://doi.org/10.1145/3025453.3025484). URL: <https://doi.org/10.1145/3025453.3025484>.
- [62] Judith S Olson, Dakuo Wang, Gary M Olson, and Jingwen Zhang. "How people write together now: Beginning the investigation with advanced undergraduates in a project course." In: *ACM Transactions on Computer-Human Interaction (TOCHI)* 24.1 (2017), pp. 1–40. DOI: [10.1145/3038919](https://doi.org/10.1145/3038919).
- [63] François Osiurak. "What Neuropsychology Tells us About Human Tool Use? The Four Constraints Theory (4CT): Mechanics, Space, Time, and Effort." In: *Neuropsychology review* 24 (Apr. 2014). DOI: [10.1007/s11065-014-9260-y](https://doi.org/10.1007/s11065-014-9260-y).
- [64] François Osiurak and Arnaud Badets. "Tool Use and Affordance: Manipulation-Based Versus Reasoning-Based Approaches." In: *Psychological review* 123 (Feb. 2016). DOI: [10.1037/rev0000027](https://doi.org/10.1037/rev0000027).
- [65] François Osiurak, Christophe Jarry, and Didier Gall. "Grasping the Affordances, Understanding the Reasoning: Toward a Dialectical Theory of Human Tool Use." In: *Psychological review* 117 (Apr. 2010), pp. 517–40. DOI: [10.1037/a0019004](https://doi.org/10.1037/a0019004).
- [66] Chris Parnin, Carsten Görg, and Spencer Rugaber. "Enriching revision history with interactions." In: *MSR '06: Proceedings of the 2006 international workshop on Mining software repositories*. May 2006. DOI: [10.1145/1137983.1138019](https://doi.org/10.1145/1137983.1138019).
- [67] Andreas S. Rath, Didier Devaurs, and Stefanie N. Lindstaedt. "UICO: An Ontology-Based User Interaction Context Model for Automatic Task Detection on the Computer Desktop." In: *Proceedings of the 1st Workshop on Context, Information and Ontologies*. CIAO '09. Heraklion, Greece: Association for Computing Machinery, 2009. ISBN: 9781605585284. DOI: [10.1145/1552262.1552270](https://doi.org/10.1145/1552262.1552270). URL: <https://doi.org/10.1145/1552262.1552270>.
- [68] Tye Rattenbury and John Canny. "CAAD: An Automatic Task Support System." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '07. San Jose, California, USA: ACM, 2007, 687–696. ISBN: 9781595935939. DOI: [10.1145/](https://doi.org/10.1145/)

- 1240624.1240731. URL: <https://doi-org.ins2i.bib.cnrs.fr/10.1145/1240624.1240731>.
- [69] Jun Rekimoto. “Time-machine computing: a time-centric approach for the information environment.” In: *Proceedings of the 12th annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA: ACM, 1999, pp. 45–54. DOI: [10.1145/320719.322582](https://doi.org/10.1145/320719.322582).
- [70] Miguel A. Renom, Baptiste Caramiaux, and Michel Beaudouin-Lafon. “Exploring Technical Reasoning in Digital Tool Use.” In: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. CHI ’22. New Orleans, LA, USA: Association for Computing Machinery, 2022. ISBN: 9781450391573. DOI: [10.1145/3491102.3501877](https://doi.org/10.1145/3491102.3501877). URL: <https://doi.org/10.1145/3491102.3501877>.
- [71] Daniela K. Rosner, Miwa Ikemiya, Diana Kim, and Kristin Koch. “Designing with Traces.” In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’13. Paris, France: ACM, 2013, 1649–1658. ISBN: 9781450318990. DOI: [10.1145/2470654.2466218](https://doi.org/10.1145/2470654.2466218). URL: <https://doi.org/10.1145/2470654.2466218>.
- [72] Chiara Rossitto, Cristian Bogdan, and Kerstin Eklundh. “Understanding Constellations of Technologies in Use in a Collaborative Nomadic Setting.” In: *Computer Supported Cooperative Work (CSCW) 23* (Apr. 2014). DOI: [10.1007/s10606-013-9196-4](https://doi.org/10.1007/s10606-013-9196-4).
- [73] Adam Rule, Aurélien Tabard, and Jim Hollan. “Using visual histories to reconstruct the mental context of suspended activities.” In: *Human–Computer Interaction 32.5-6* (2017), pp. 511–558. DOI: [10.1080/07370024.2017.1300063](https://doi.org/10.1080/07370024.2017.1300063).
- [74] Benedikt Schmidt, Sebastian Doeweling, and Max Mühlhäuser. “Interaction History Visualization.” In: *Proceedings of the 30th ACM International Conference on Design of Communication*. SIGDOC ’12. Seattle, Washington, USA: ACM, 2012, 261–270. ISBN: 9781450314978. DOI: [10.1145/2379057.2379107](https://doi-org.ins2i.bib.cnrs.fr/10.1145/2379057.2379107). URL: <https://doi-org.ins2i.bib.cnrs.fr/10.1145/2379057.2379107>.
- [75] Benedikt Schmidt and Eicke Godehardt. “Interaction Data Management.” In: *Proceedings of the 15th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems - Volume Part II*. KES’11. Kaiserslautern, Germany: Springer-Verlag, 2011, 402–409. ISBN: 9783642238628.
- [76] Benedikt Schmidt, Eicke Godehardt, and Björn Pantel. “Visualizing the work process: situation awareness for the knowledge worker.” In: (2012).

- [77] Benedikt Schmidt, Johannes Kastl, Todor Stoitsev, and Max Mühlhäuser. "Hierarchical Task Instance Mining in Interaction Histories." In: *Proceedings of the 29th ACM International Conference on Design of Communication*. SIGDOC '11. Pisa, Italy: Association for Computing Machinery, 2011, 99–106. ISBN: 9781450309363. DOI: [10.1145/2038476.2038495](https://doi.org/10.1145/2038476.2038495). URL: <https://doi.org/10.1145/2038476.2038495>.
- [78] Victoria Schwanda Sosik, Xuan Zhao, and Dan Cosley. "See friendship, sort of: How conversation and digital traces might support reflection on friendships." In: *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*. New York, NY, USA: ACM, 2012, pp. 1145–1154. DOI: [10.1145/2145204.2145374](https://doi.org/10.1145/2145204.2145374).
- [79] Jianqiang Shen, Lida Li, and Thomas Dietterich. "Real-Time Detection of Task Switches of Desktop Users." In: *International Joint Conference on Artificial Intelligence*. Jan. 2007, pp. 2868–2873.
- [80] B. Shneiderman. "Direct Manipulation: A Step Beyond Programming Languages." In: *Computer* 16.8 (1983), 57–69. ISSN: 0018-9162. DOI: [10.1109/MC.1983.1654471](https://doi.org/10.1109/MC.1983.1654471). URL: <https://doi.org/10.1109/MC.1983.1654471>.
- [81] David Smith, Charles Irby, Ralph Kimball, and Eric Harslem. "The Star User Interface: An Overview." In: June 1982, pp. 515–528. ISBN: 978-1-4419-2881-8. DOI: [10.1145/1500774.1500840](https://doi.org/10.1145/1500774.1500840).
- [82] David Smith, Charles Irby, Ralph Kimball, Bill Verplank, and Eric Harslem. "Designing the Star user interface." In: *Xerox Star Project* (Jan. 1982), pp. 261–283.
- [83] Ralph Taylor. "Human Territorial Functioning: An Empirical, Evolutionary Perspective on Individual and Small Group Territorial Cognitions, Behaviors, and Consequences." In: (Aug. 1988). DOI: [10.1017/CB09780511571237](https://doi.org/10.1017/CB09780511571237).
- [84] Manuela Waldner, Stefan Bruckner, and Ivan Viola. "Graphical histories of information foraging." In: *Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational*. New York, NY, USA: ACM, 2014, pp. 295–304. DOI: [10.1145/2639189.2641202](https://doi.org/10.1145/2639189.2641202).
- [85] Dakuo Wang, Judith S Olson, Jingwen Zhang, Trung Nguyen, and Gary M Olson. "DocuViz: visualizing collaborative writing." In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2015, pp. 1865–1874. DOI: [10.1145/2702123.2702517](https://doi.org/10.1145/2702123.2702517).
- [86] Dakuo Wang, Haodan Tan, and Tun Lu. "Why Users Do Not Want to Write Together When They Are Writing Together: Users' Rationales for Today's Collaborative Writing Practices." In: *Pro-*

- ceedings of the ACM on Human-Computer Interaction* 1 (Dec. 2017), pp. 1–18. DOI: [10.1145/3134742](https://doi.org/10.1145/3134742).
- [87] Karl E Weick, Kathleen M Sutcliffe, and David Obstfeld. “Organizing and the process of sensemaking.” In: *Organization science* 16.4 (2005), pp. 409–421. DOI: [10.1515/9783038212843.216](https://doi.org/10.1515/9783038212843.216).
- [88] Sungjoon Steve Won, Jing Jin, and Jason I Hong. “Contextual web history: using visual and contextual cues to improve web browser history.” In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2009, pp. 1457–1466. DOI: [10.1145/1518701.1518922](https://doi.org/10.1145/1518701.1518922).
- [89] Krist Wongsuphasawat, John Alexis Guerra Gómez, Catherine Plaisant, Taowei David Wang, Meirav Taieb-Maimon, and Ben Shneiderman. “LifeFlow: visualizing an overview of event sequences.” In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2011, pp. 1747–1756. DOI: [10.1145/1978942.1979196](https://doi.org/10.1145/1978942.1979196).
- [90] Leila Yahiaoui, Yannick Prié, and Zizette Boufaïda. “The Redocumentation Process of Computer Mediated Activity Traces: A General Framework.” In: *Proceedings of the 20th ACM Conference on Hypertext and Hypermedia*. HT '09. Torino, Italy: ACM, 2009, 363–364. ISBN: 9781605584867. DOI: [10.1145/1557914.1557990](https://doi.org/10.1145/1557914.1557990). URL: <https://doi-org.ins2i.bib.cnrs.fr/10.1145/1557914.1557990>.



APPENDIX

A.1 PRIVACY CONCERNS AND SHARED HISTORIES QUESTIONNAIRES

Introduction:

As a reminder, an activity trace is a piece of information about past actions on the project, which will be left by you or one of your team members.

It will allow you in some cases to keep track of the progress on the project.

It can be, for example, different versions of a document, annotations or comments, modification histories, discussion/communication histories, or even written notes.

It can be data shared between the different actors in the project, or data that you keep personally.

Privacy concerns questionnaire

The following statements are related to your experience, or your feelings towards real-time collaborative work tools (such as Google Doc).

During a collaborative writing project:

- I would be willing to use tools to write text in real time with other members of my team.

1. strongly agree 2. agree 3. neither agree nor disagree
4. disagree 5. strongly disagree

- I would be willing to store our common work around the writing project in a shared environment.

1. strongly agree 2. agree 3. neither agree nor disagree
4. disagree 5. strongly disagree

- I would be willing to share in real-time data such as my modification history or my comments on a shared document.

1. strongly agree 2. agree 3. neither agree nor disagree
4. disagree 5. strongly disagree

- I would be willing to share with the other members of my team the history of discussions I have had around the project.

1. strongly agree 2. agree 3. neither agree nor disagree
4. disagree 5. strongly disagree

- I would be willing to share personal material such as my written notes, or my summaries of meetings about the project.

1. strongly agree 2. agree 3. neither agree nor disagree
4. disagree 5. strongly disagree

Shared histories questionnaire

The following statements are to be rated based on things you felt were missing or needed during the writing project you had in mind for the previous statements.

It would have been necessary for my current tools:

- To be able to synthesize the progress of my team on a project at key moments.

1. strongly agree 2. agree 3. neither agree nor disagree
4. disagree 5. strongly disagree

- To have a common and unique space where to gather all my team's communications about the project.

1. strongly agree 2. agree 3. neither agree nor disagree
4. disagree 5. strongly disagree

- To be able to link the different communication channels with the discussed collaborative documents.

1. strongly agree 2. agree 3. neither agree nor disagree
4. disagree 5. strongly disagree

- To be able to visualize all activity traces in a single space to be able to navigate through them.

1. strongly agree 2. agree 3. neither agree nor disagree
4. disagree 5. strongly disagree

- To be able to link my personal traces (written notes, recordings, non-shared documents...) with my collaborative project.

1. strongly agree 2. agree 3. neither agree nor disagree
4. disagree 5. strongly disagree

A.2 TRACELENS STUDY QUESTIONNAIRES

Participant ID :

Pre-study questionnaire

- I happen to face issues in differentiating traces within the same history, but conveying different kinds of information.

For example differentiating annotations or communications depending on their topic.

1- strongly disagree 2- disagree 3- neither agree nor disagree
4- agree 5- strongly agree

- I happen to face issues in regrouping traces left within different histories, but conveying the same kind of information.

For example regrouping annotations, messages, or mails that are all about the same topic.

1- strongly disagree 2- disagree 3- neither agree nor disagree
4- agree 5- strongly agree

- Overall, I have a hard time navigating interaction and communication histories.

1- strongly disagree 2- disagree 3- neither agree nor disagree
4- agree 5- strongly agree

- I happen to face issues in communicating my progress to my team.

1- strongly disagree 2- disagree 3- neither agree nor disagree
4- agree 5- strongly agree

- I happen to face difficulties sharing important traces with my team.

1- strongly disagree 2- disagree 3- neither agree nor disagree
4- agree 5- strongly agree

- I happen to face issues in understanding my team's progress.

1- strongly disagree 2- disagree 3- neither agree nor disagree
4- agree 5- strongly agree

- I happen to face issues recalling how I previously found information in some of my histories.

1- strongly disagree 2- disagree 3- neither agree nor disagree
4- agree 5- strongly agree

- Overall, I have a hard time following my team's progress in a collaborative project.

1- strongly disagree 2- disagree 3- neither agree nor disagree
4- agree 5- strongly agree

Participant ID :

Post-study questionnaire

- This kind of system would improve differentiating traces within the same history, but conveying different information.
1- strongly disagree 2- disagree 3- neither agree nor disagree
4- agree 5- strongly agree

- This kind of system would improve regrouping traces left within different histories, but conveying similar information.
1- strongly disagree 2- disagree 3- neither agree nor disagree
4- agree 5- strongly agree

- Overall, this kind of system where histories are shared across applications would help me navigate in interaction and communication histories.
1- strongly disagree 2- disagree 3- neither agree nor disagree
4- agree 5- strongly agree

- In a collaborative project, this kind of system would improve my way to communicate my progress to my team.
1- strongly disagree 2- disagree 3- neither agree nor disagree
4- agree 5- strongly agree

- With this kind of system, it would be easier to share important traces with my team.
1- strongly disagree 2- disagree 3- neither agree nor disagree
4- agree 5- strongly agree

- In a collaborative project, this kind of system would improve my understanding of my team's progress.
1- strongly disagree 2- disagree 3- neither agree nor disagree
4- agree 5- strongly agree

- With this kind of system, it would be easier to recall how I previously found information in some of my histories.
1- strongly disagree 2- disagree 3- neither agree nor disagree
4- agree 5- strongly agree

- Overall, this kind of system where histories are shared across applications would be beneficial to follow my team's progress in a collaborative project.
1- strongly disagree 2- disagree 3- neither agree nor disagree
4- agree 5- strongly agree

Participant ID :

Additional information

Age :

- 18 - 25
- 25 - 30
- 30 - 35
- 35 - 40
- 40 - 45
- 45 - 50
- 55 - 60
- more than 60
- do not want to answer

Gender :

- male
- female
- not listed :
- do not want to answer

A.3 TRACEPICKER STUDY QUESTIONNAIRES

Participant ID :

Pre-study questionnaire

- Q1: With my current tools' histories, I feel in control over what information my histories display.
For example, searching for a specific item/filtering actions.
1- strongly disagree 2- disagree 3- neither agree nor disagree
4- agree 5- strongly agree
- Q2: With my current tools' histories, it is easy to deal with a lot of history information.
1- strongly disagree 2- disagree 3- neither agree nor disagree
4- agree 5- strongly agree
- Q3: With my current tools' histories, it is easy to contextualize my own past actions.
Contextualize means associate meaning with actions. For example, you modified your conclusion because your supervisor told you to.
1- strongly disagree 2- disagree 3- neither agree nor disagree
4- agree 5- strongly agree
- Q4: With my current tools' histories, it is easy to contextualize my team's past actions.
1- strongly disagree 2- disagree 3- neither agree nor disagree
4- agree 5- strongly agree
- Q5: With my current tools' histories, it is easy to resume work after a short break.
After a lunch break, a day off.
1- strongly disagree 2- disagree 3- neither agree nor disagree
4- agree 5- strongly agree
- Q6: With my current tools' histories, it is easy to resume work after a long term
After one or two weeks off.
1- strongly disagree 2- disagree 3- neither agree nor disagree
4- agree 5- strongly agree
- Q7: Overall, it is easy to resume working with my current tools' histories.
1- strongly disagree 2- disagree 3- neither agree nor disagree
4- agree 5- strongly agree

Participant ID :

Additional information

Age :

- 18 - 25
- 25 - 30
- 30 - 35
- 35 - 40
- 40 - 45
- 45 - 50
- 55 - 60
- more than 60
- do not want to answer

Gender :

- male
- female
- not listed :
- do not want to answer