Designing Pen-based Interactions for Productivity and Creativity

Vincent Cavez

THESE DE DOCTORAT

NNT : 2025UPASG013

université PARIS-SACLAY

Designing Pen-based Interactions for Productivity and Creativity

Concevoir des Interactions au Stylet pour la Productivité et la Créativité

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 **Emmanuel PIETRIGA**, Directeur de Recherche, et le co-encadrement de **Caroline APPERT**, Directrice de Recherche.

Thèse soutenue à Paris-Saclay, le 3 mars 2025, par

Vincent CAVEZ

Composition du jury

Membres du jury avec voix délibérative

Petra ISENBERG Directrice de recherche, Inria Saclay	Présidente
Daniel VOGEL Professor, University of Waterloo	Rapporteur & Examinateur
Marcelo WANDERLEY Professor, McGill University	Rapporteur & Examinateur
Hariharan SUBRAMONYAM Research Assistant Professor, Stanford University	Examinateur
Theophanis TSANDILAS Chargé de recherche, Inria Saclay	Examinateur

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

Titre: Concevoir des Interactions au Stylet pour la Productivité et la Créativité
Mots clés: Interaction au stylet, Interaction tactile, Manipulation interactive de données, Tableurs, Composition musicale, Intelligence artificielle

Résumé: Conçus pour une utilisation avec la souris et le clavier, les outils aidant à la productivité et à la créativité sont puissants sur les ordinateurs de bureau, mais leur structure devient un obstacle lorsqu'ils sont transposés sur des surfaces interactives offrant une saisie tactile et au stylet.

En effet, les opportunités offertes par le stylet en termes de précision et d'expressivité ont été démontrées dans la littérature sur en IHM. Cependant, les outils de productivité et de créativité nécessitent une refonte minutieuse exploitant ces propriétés uniques pour tirer parti de l'intuitivité qu'ils offrent, tout en conservant les avantages liés à la structure. Cette articulation délicate entre le stylet et la structure a été négligée dans la littérature.

Mon travail de thèse se concentre sur cette articulation à travers deux cas d'utilisation afin de répondre à la question de recherche générale : *« Comment concevoir des interactions au stylet pour la productivité et la créativité sur des surfaces interactives ? »* Je considère que la productivité dépend de l'efficacité, tandis que la créativité repose à la fois sur l'efficacité et la flexibilité, et j'explore des interactions qui favorisent ces deux dimensions.

Mon premier projet, *EunomInk*, explore un ensemble de techniques d'interaction basées sur le stylet et conçues pour les logiciels de tableurs, et propose des lignes directrices pour promouvoir l'efficacité sur les surfaces interactives. Je commence par analyser les logiciels commerciaux et par mener une étude d'élicitation pour comprendre ce que les utilisateurs peuvent faire et ce qu'ils aimeraient faire avec les tableurs sur des surfaces interactives. Sur la base de ces analyses, je conçois des techniques d'interaction qui exploitent les opportunités offertes par le stylet pour réduire les frictions et permettre plus d'opérations par manipulation directe sur et à travers la grille. Je prototype ces techniques d'interaction et mène une étude qualitative auprès d'utilisateurs qui effectuent diverses opérations sur tableurs avec leurs propres données. Les observations montrent que l'utilisation du stylet pour contourner la structure constitue un moyen prometteur de favoriser l'efficacité dans un outil de productivité.

Mon deuxième projet, *EuterPen*, explore un ensemble de techniques d'interaction basées sur le stylet, et conçues pour les logiciels de notation musicale, et propose des lignes directrices pour promouvoir à la fois l'efficacité et la flexibilité sur les surfaces interactives. Je commence par une série de neuf entretiens avec des compositeurs professionnels afin de prendre du recul et de comprendre à la fois leur processus de réflexion et leur processus de travail avec leurs outils actuels sur ordinateur de bureau. Sur la base de cette analyse double, j'élabore des lignes directrices pour la conception de fonctionnalités ayant le potentiel de promouvoir à la fois l'efficacité pour les opérations fréquentes ou complexes et la flexibilité dans l'exploration des idées. Ensuite, je mets en œuvre ces lignes directrices à travers un processus de conception itératif : deux phases de prototypage, un atelier de conception participative et une série finale d'entretiens avec huit compositeurs professionnels. Les observations montrent qu'en plus d'utiliser le stylet pour profiter de la structure afin de favoriser l'efficacité, tirer parti de ses propriétés pour briser temporairement la structure constitue un moyen prometteur de promouvoir la flexibilité dans un outil de soutien à la créativité.

Je conclus ce manuscrit en discutant de différentes manières d'interagir avec la structure, en présentant un ensemble de recommandations pour soutenir la conception d'interactions basées sur le stylet pour les outils de productivité et de créativité, et en élaborant sur les applications futures que cette thèse ouvre.

Title: Designing Pen-based Interactions for Productivity and Creativity
Keywords: Touch interaction, Pen interaction, Interactive data manipulation, Spreadsheets, Music composition, Artificial intelligence

Abstract: Designed with the mouse and keyboard in mind, productivity tools and creativity support tools are powerful on desktop computers, but their structure becomes an obstacle when brought to interactive surfaces supporting pen and touch input.

Indeed, the opportunities provided by the pen for precision and expressivity have been demonstrated in the HCI literature, but productivity and creativity tools require a careful redesign leveraging these unique affordances to take benefit from the intuitiveness they offer while keeping the advantages of structure. This delicate articulation between pen and structure has been overlooked in the literature.

My thesis work focuses on this articulation with two use cases to answer the broad research question: *"How to design pen-based interactions for productivity and creativity on interactive surfaces?"* I argue that productivity depends on efficiency while creativity depends on both efficiency and flexibility, and explore interactions that promote these two dimensions.

My first project, *EunomInk*, explores a set of pen-based interaction techniques designed for spreadsheet programs and contributes guidelines to promote efficiency on interactive surfaces. I first conduct an analysis of commercial spreadsheet programs and an elicitation study to understand what users can do and what they would like to do with spreadsheets on interactive surfaces. Informed by these, I design interaction techniques that leverage the opportunities of the pen to mitigate friction and enable more operations by direct manipulation on and through the grid. I prototype these interaction techniques and conduct a qualitative study with information workers who performed a variety of spreadsheet operations on their own data. The observations show that using the pen to bypass the structure is a promising mean to promote efficiency with a productivity tool.

My second project, *EuterPen*, explores a set of pen-based interaction techniques designed for music notation programs and contributes guidelines to promote both efficiency and flexibility on interactive surfaces. I first conduct a series of nine interviews with professional composers in order to take a step back and understand both their thought process and their work process with their current desktop tools. Building on this dual analysis, I derive guidelines for the design of features which have the potential to promote both efficiency with frequent or complex operations and flexibility in regard to the exploration of ideas. Then, I act on these guidelines by engaging in an iterative design process for interaction techniques that leverage the opportunities of the pen: two prototyping phases, a participatory design workshop, and a final series of interviews with eight professional composers. The observations show that on top of using the pen to leverage the structure for efficiency, using its properties to temporarily break the structure is a promising mean to promote flexibility with a creativity support tool.

I conclude this manuscript by discussing several ways to interact with structure, presenting a set of guidelines to support the design of pen-based interactions for productivity and creativity tools, and elaborating on the future applications this thesis opens.

Acknowledgments

J'aimerais tout d'abord remercier mes parents pour m'avoir donné la passion que j'ai pour les sciences et la musique, ainsi que la confiance et la structure dont j'avais besoin pour mener à bien mes études.

J'aimerais également remercier tous mes amis, les plus proches comme les plus éloignés, que ce soit dans l'espace ou dans le temps, car vous avez tous, à votre manière, joué un rôle indispensable dans ma vie, et dans la construction de la personne que je suis aujourd'hui. Beaucoup d'entre vous ont même eu un rôle direct dans la réalisation de cette thèse, en acceptant volontier de participer à mes études et en m'encourageant dans mes recherches. Un merci tout particulier à Lucius Arkmann pour son implication dans la totalité de mon second projet, EuterPen.

J'aimerais remercier tous les professeurs et chargés de TD qui ont vu du potentiel en moi, et m'ont fourni les clefs et l'accompagnement nécessaires à mon parcours.

J'aimerais remercier mes collègues lors de mon stage de Master, Oscar et Antonin, qui m'ont soutenu avec mon premier projet d'Interaction Humain-Machine et m'ont encouragé à poursuivre une thèse de doctorat.

J'aimerais remercier très chaleureusement Emmanuel et Caroline, pour être les encadrants incroyables qu'ils sont. Pendant trois ans et trois mois, ils ont toujours été à l'écoute, gentils, drôles, et évidemment brillants. Ils m'ont fourni un environnement de travail idéal, et je ne peux imaginer une meilleure manière dont cette thèse aurait pu se passer.

Bien sûr, j'aimerais remercier tout le reste de l'équipe ainsi que mes amis chez Aviz et Exsitu, pour les nombreux conseils et coups de main que vous m'avez donné, mais aussi pour avoir contribué à rendre cette thèse si agréable.

Un grand merci à ma fiancée, Lucie, pour son soutien infaillible pendant cette épreuve et pour la confiance qu'elle place en moi et en mes divers projets.

Et finalement, j'aimerais remercier les membres de mon jury de thèse, Daniel, Marcelo, Hari, Petra and Theophanis. Merci pour votre présence et l'intérêt que vous avez porté à mes travaux. C'est un vrai plaisir d'avoir pu partager avec vous toutes ces idées et réflexions sur mes recherches passées, mais aussi celles à venir. Sans vous, tout cela n'aurait pas été possible.

Encore merci à tous !

Contents

Acknowledgments iii

List of Figures vii

List of Tables xii

1 Introduction 1
- 1.1 Productivity and Creativity 1
- 1.2 Tools built for Desktop Computers... 2
- 1.3 ... and brought to Interactive Surfaces 3
- 1.4 Opportunities of the Digital Pen 5
- 1.5 Research questions 8
- 1.6 Thesis outline 9

2 Related Work 11
- 2.1 Interacting with Pen and Touch 11
- 2.2 Interacting with Spreadsheets 14
- 2.3 Interacting with Music Scores 17

3 Bypassing the Structure for Productivity 25
- 3.1 Motivation 25
- 3.2 What Users *Can* Do with Commercial Spreadsheet Programs 27
- 3.3 What Users *Could* Do with Spreadsheet Programs 30
  - 3.3.1 Participants 31
  - 3.3.2 Apparatus 31
  - 3.3.3 Task and Procedure 31
  - 3.3.4 Results 33
- 3.4 EunomInk: a Spreadsheet Program Prototype 37
  - 3.4.1 Mitigating Friction Between the Grid and Value Layers 38
  - 3.4.2 Mitigating Friction Between the Grid and Navigation Layers 43
- 3.5 Qualitative Study 46
  - 3.5.1 Participants and Apparatus 48
  - 3.5.2 Task and Procedure 48
  - 3.5.3 Results 49

3.6 Conclusion . . . 56
3.7 Future Work . . . 57

4 Breaking the Structure for Creativity **59**
4.1 Motivation . . . 60
4.2 How Composers Think and Work . . . 61
4.2.1 Insights on Thought Process . . . 64
4.2.2 Analysis of Work Process . . . 67
4.3 Design Opportunities . . . 79
4.3.1 Breaking Down Musical Elements . . . 80
4.3.2 Breaking the Score's Homogeneity . . . 81
4.3.3 Breaking the Score's Linear Structure . . . 83
4.3.4 Discussion . . . 84
4.4 EuterPen: A Music Notation Program Prototype . . . 85
4.4.1 Design Journey . . . 86
4.4.2 Software Implementation . . . 88
4.4.3 Design Principles . . . 88
4.4.4 Breaking the Score's Linear Structure . . . 89
4.4.5 Breaking the Score's Homogeneity . . . 94
4.4.6 Breaking Down Musical Elements . . . 99
4.5 Discussion . . . 104
4.6 Conclusion . . . 106
4.7 Future Work . . . 107

5 Conclusion and Perspectives **109**
5.1 Summary of Contributions . . . 109
5.2 Discussion . . . 111
5.3 Key Insights for Future Work . . . 112
5.4 Examples of Applications . . . 113
5.5 Final Word . . . 116

A Appendices **145**
A.1 Chapter 3: Elicitation Study . . . 145
A.1.1 List of referents . . . 145
A.1.2 Definition of a Sign . . . 145
A.2 Chapter 3: Implementation Details . . . 146
A.2.1 Prototype Implementation . . . 146
A.2.2 Generalizing Subcell Selections . . . 146

List of Figures

1.1 Microsoft Excel and Dorico [44] WIMP interfaces on iPad. . . . 4

1.2 Left: Apple Newton MessagePad 110 (1994) [47] Middle: Blackberry Bold 9000 (2008) [66] Right: Apple iPad Pro (2015) [7] . . . 5

2.1 Left: **Tableur** [203] lets users sketch values by hand on a blank canvas, segment them into a table which can then be edited. Top right: **WritLarge** [195] lets users seamlessly switch between a formatted table (a) and a hand-drawn grid to edit the strokes (b). Handwriting on a formatted table is immediately recognized as text (c). Bottom right: **Style Blink** [149] lets users create simple tables by sketching and interact with the strokes to move content. 15

2.2 **TouchPivot** [85]. (a) Users can filter out a category by striking out the category on the table or visualizations. (b) Users can change the columns in a pivot transformation by tapping on the names in a chart title and selecting other columns. (c) Users can draw a lasso on a scatterplot to see details on the points contained in the lasso. Similarly, they can draw a circle on the table to locate the selected records on the scatterplot. 16

2.3 **Presto2**. An extract from the table of gestures presented in [119]. 20

2.4 **Mus*ink*** [179]. On the left, a scenario: (a) Drawing on paper: expressing a new type of crescendo (b) OpenMusic: defining the crescendo's vibration pattern (c) Mus*ink* Gesture Browser: defining the crescendo class. On the right, (d) a score created by a composer during an interview is shown on the Gesture Browser (e) a representation of Mus*ink* gestures linked with detailed graphs, supporting functions over curves. 21

2.5 At the top left, **Paper Substrates** [57] used to edit a musical sequence on a printed interface with physical data (blue) and on the Max/MSP interface with digital data (yellow). At the bottom left, **PaperTonnetz** [55] used to create and listen to a musical sequence drawn on a two-dimensional heptatonic layout printed on interactive paper. On the right, **PaperComposer** [56], a graphical interface builder that allows users to create a personalized interactive paper interface that they can connect to their own computer-based musical data. 22

3.1 The three layers of a spreadsheet program's user interface on an interactive surface. 26

3.2 Sample configurations used in the analysis of commercial spreadsheet programs: *Numbers* running on an Apple iPad Pro; *Excel* running on a Microsoft Surface Book; *Google Sheets* running on a Wacom Cintiq Pro. 28

3.3 Four different actions triggered when dragging with a single finger on an already-selected cell: (a) pan the sheet P; (b) cut & paste the content of the cells from the current selection GM_{cp}; (c) reset the selection and then adjust it GS_n,GS_a; (d) adjust the current selection without resetting it GS_a. 30

3.4 Three different behaviors when performing a pen double-tap between characters `o` and `k` in cell `Tokyo`: enter value-edit mode, and either (a) position the caret at the end of the string (VS_{lc}); (b) position the caret in the string, precisely between those two characters (VS_{nc}); or (c) select the whole string (VS_{ac}). 30

3.5 A trial in the study. The *referent* is presented as a question in the top panel. The left and right spreadsheets respectively show the state *before* and *after* the referent's effect is applied. Participants use their fingers and pen directly on the left panel to make their *sign* proposal. They then evaluate their proposal with a *Match* and *Easy* score using the 5-pt Likert scales displayed below the spreadsheets. 32

3.6 (a) Agreement about *sign* proposals per referent. (b) Agreement about *modality* proposals per referent. 34

3.7 (a) Distribution of Match scores per referent. (b) Distribution of Easiness scores per referent. 36

3.8 Performing a subcell selection by drawing an arbitrary mark: (a) the pen leaves ink, providing an indication of what substring will be selected; (b) once the pen lifted, the ink is replaced by a beautified representation of the selection, consisting of the substring actually delineated (dark blue) and its complements on either or both sides (light blue). 38

3.9 Removing a substring across all cells of a column that contains the surname, title and first name of people embarked on the Titanic: (a) circling the end of one cell value using the pen, from the comma separating the title from the first name, to the end of the string; (b) dragging with the pen from the resulting subcell selection to the column header to generalize it to all values in the column; (c) tapping with the pen eraser on any of those subcell selections deletes them all, leaving only the surname and title (d). 40

3.10 Splitting title and surname in two separate columns. After generalization of the person's title selection (from comma to end of string), (a) performing an outward pinch gesture with one finger on the main subcell selection and another finger on the complement splits the two in separate columns (b). 41

3.11 Inverting first name and surname (`John Smith` → `Smith, John`) by direct value-level manipulation in one cell: (a) dragging the surname subcell selection to the left; (b-c) inserting a comma followed by a white space between surname and first name. . 42

3.12 Value-level editing. (a) Entering value-edit mode by performing a vertical pen drag gesture within the cell's boundaries; (b) handwriting the new cell value which gets recognized and (c) inserted in the cell. 43

3.13 Minitable widget for pen selections in large tables. a) Selecting from rows 25 to 39 using the minitable. A preview of the row that falls below the cursor (row 39 here) is visible left of the pen tip. Row selection start and end indices are also displayed. b) Brushing through columns in the gutter above the table shows a preview of the columns (column L here, as indicated next to the pen tip, showing where passengers embarked). Light blue lines in the minitable give an overview of which rows are currently selected (in this example, all rows with Pclass=2). 44

3.14 Minivis plots. (a) A bar chart shows the distribution of values for categorical variables – here one of three ports where passengers could embark on the Titanic. The blue bar indicates that the user has selected rows featuring C (for *Cherbourg*) as the port of embarkation. (b) A density plot and a box plot are juxtaposed to visualize the distribution for quantitative variables – here the age of passengers. The blue area in the box plot indicates that the user has selected rows featuring an Age value that falls in the third quartile. 45
3.15 EunomInk, the prototype Web-based application used in the qualitative study. Several features are illustrated in this screen capture: (a) minivis plots for row selection; (b) minitable for grid selection and worksheet navigation; (c) substring selection across cells; (d) annotation palette; (e) freeform ink annotation; (f) post-it annotation; (g) post-it annotation minimized. 47
3.16 (a) Interaction sequences of the think-aloud phase, with one timeline track per participant. (b) Relative time spent performing interactions of different types, per participant. 50
3.17 Participants' evaluation of (a) usefulness and (b) ease-of-use for 18 interactive features using 5-point Likert scales. 51
3.18 Inverting first name and surname (`John Smith` → `Smith, John`) by direct value-level manipulation as in Figure 3.11, but for all cells at once: (a) dragging the generalized surname subcell selection to the left; (b-c) inserting a comma followed by a white space between all surnames and first names. 54
3.19 Our techniques enable seamless direct manipulations of diverse elements in a spreadsheet primarily by enabling a broader variety of selection actions than existing spreadsheet programs. With only one or two pen marks, users can select different types of scope. Representative selections at the grid and value levels: a) a single cell; b) large sets of cells without navigation – here columns `B-J` extending far beyond the current viewport; c) a substring within a cell; d) multiple substrings matching a pattern across cells in a column. 56

4.1 Methodology for analyzing raw interview transcripts. 63
4.2 (a) Musical notations, textual annotations and sketches organized in a sorter. (b) Dialog window for the creation of a new abstraction. (c) Hidden bar in a passage that does not need the rhythmic frame of a measure. 69
4.3 (a) Dialog box used to specify properties to copy. (b) Vertical menu to store ideas aside from the score. (c) Written orchestration ideas on a printed score. 71

4.4 Mock-up illustrations of design opportunities: (a) using the pen to sketch a rhythmic pattern to be searched in the score, differentiating between perfect (yellow) and partial (orange) matches; (b) inserting heterogeneous content in-context: handwritten symbols that do not get automatically beautified, and a playable audio recording represented using its spectrum; (c) selecting (blue stroke) and copying measures to a canvas where passages remain editable but can be laid out freely, ignoring layout rules imposed by the staff notation. 82

4.5 EuterPen running on a desktop interactive surface. Areas around staves (called pensieves) are used to freely explore musical ideas while keeping the main score stable, and to store content that supports composers' creative process: documents, audio samples, diagrams. Handwritten and engraved music notations can coexist everywhere, and are both amenable to structured-yet-flexible interactive manipulations, including a tool to search for constrained or relaxed melodic patterns across the whole workspace. 86

4.6 Carving a pensieve by (a-b) stretching space between two staves using a simple multitouch gesture, and then (c) populating that space with multiple objects using drag-&-drop. An audio sample has been dropped already, and the user is about to place an empty staff to write music on. 90

4.7 Breaking the score's linear structure: participants' evaluation of the features demonstrated in the videos shown during the interviews. 92

4.8 A note selection and the associated contextual menu: a) erase selection, b-c) copy-&-paste selection, d) play back selection, e) find occurrences of the selection, f/f') switch between engraved and handwritten representations. 95

4.9 Breaking the score's homogeneity: participants' evaluation of the features demonstrated in the videos shown during the interviews. 97

4.10 The three predefined pen gestures available to composers, that coexist with all direct manipulations on the engraved notation as well as the input of handwritten notation – all without any mode-switch : (a) lasso-to-select; (b) pin-to-play; (c) caret-to-insert (the amount of space to insert being controlled by the length of the upward trace, here 9 bars). 99

4.11 Searching for all occurrences of a melodic pattern based on pitch only, relaxing constraints on rhythm. (a) Four note heads have been selected (colored blue, 7th staff), and all occurrences of the same pitch sequence (C-B-C-D) are highlighted yellow, regardless of their duration. Occurrences of the same sequence but transposed are also highlighted (*e.g.*, E♭-D-E♭-F, 11th staff). (b) The sixteen occurrences are also highlighted on the interactive minimap that gives an overview of the whole score. Annotations made on the score are also visible on the minimap, providing composers with navigation landmarks. . . 101
4.12 Breaking down musical elements: participants' evaluation of the features demonstrated in the videos shown during the interviews. 102

5.1 A continuum of structural interaction in content editing programs. 112

List of Tables

3.1 Summary of user actions and their effect, by spreadsheet program, by operating system, by display surface hardware. Actions are grouped in four categories, which are visually encoded with redundancy to help identify them based on the action code's first letter and background color: grid-level selection GS_* and manipulation GM_*; value-level selection VS_*; P, PZ panning & zooming the worksheet; MN invoking contextual widgets. Empty cells indicate that this input has no effect. 29

4.1 Profile of the nine composers interviewed. *Activities: C=Composition; A=Arrangement; E=Engraving for another composer. 62
4.2 The list of cognitive dimensions, adapted from Nash [118] and Whitley & Blackwell [185], that we used in our analysis. 68
4.3 Profile of the eight composers we worked with. *Activities: C=Composition; A=Arrangement; E=Engraving for another composer; T=Teaching. All of them participated in the final interviews and two of them also participated in the design workshop (we refer to them as C_1 and C_2 for anonymity concerns). 85

A.1 Referents considered in the elicitation study. 149

Chapter 1

Introduction

People aiming for productivity or creativity usually rely on the use of digital tools to help them achieve their goals during part of their process or for the entirety of if. These tools are structured to support the operations users need (*e.g.*, copy-&-paste, transform, filter), and do so by enforcing rules on the document and displaying an interface with menus, side panels and layers. Although powerful on desktop computers equipped with mouse and keyboard, their structure becomes an obstacle when brought to other devices with different characteristics. Interactive surfaces which support pen and touch input in particular, require a careful redesign leveraging their unique affordances to take benefit from the intuitiveness they offer while keeping the advantages of structure. Such a redesign even has the potential to uncover new opportunities for productive and creative work. In this thesis, I will explore how pen-based interactions can be better articulated with the fundamental structure of these digital tools, in the concrete contexts of productive work on spreadsheets and creative work on music scores.

1.1 Productivity and Creativity

Productivity has been widely studied in various disciplines such as Economics, Business, Sociology, Management sciences, Operations research, and Public administration [34]. Productivity definitions vary but all have in common that productivity is a **rate**. This rate is usually described in the scientific literature [87, 51] as an output (product or service) over an input (time, energy, materials, people). The lesser the input for the same output, the greater the productivity. In other terms, this rate depends on **efficiency**. From a *Human-Computer Interaction* point of view, this means enabling users to perform the right tasks and in the best possible way, reducing the quantity of wasted time and energy when working to produce something:

Productivity

the ability of a user to produce value efficiently

Unlike Productivity, Creativity is a concept much more complex to grasp and has been subject to debate in the Psychology and Education literature since the 1950s [68, 143]. While dictionaries still fail to produce a consistent definition [140, 115, 24, 43], the scientific literature nowadays seems to consider the following as the definition of creativity [20, 192, 167, 153, 200, 177, 168]:

Creativity

the ability of a user to produce novelty and value

Dissimilar to straightforward contexts where the user is only aiming for efficiency to produce value (filling out a spreadsheet, writing a report, preparing a presentation), the context of creative work adds another requirement: novelty. In order to be imaginative and inventive, the user needs the **ability** to explore, try, reflect, throw aside, and repeat this process, often relying on divergent thinking [152], while still generating something meaningful and relevant within its context. *Ergo*, Creativity can be seen as "Creative productivity": while still needing **efficiency** to meet their goals, users now additionally need **flexibility** [1, 121, 113, 162]. Whether we consider the case of professionals working for a living, of children studying in school, or of people engaging in activities for leisure, the following can be applied:

> *If you have something to show for, you have been productive.*
> *If you have something new to show for, you have been creative.*

1.2 Tools built for Desktop Computers...

Software supporting productivity and creativity was first developed for desktop computers in the 1980s and early 1990s, and quickly became accessible to a wide range of users thanks to their comprehensible graphical user interfaces (GUIs). The desktop setup — equipped with a keyboard, mouse, and relatively large display — led developers to create feature-rich, structured interfaces tailored to it. Early software applications, leveraging the computer's ability to support complex tasks, were well-suited to professionals and students in many fields and quickly became adopted by the majority of users.

These structured "WIMP" GUIs [41] - Windows, Icons, Menus, Pointer - not only defined user experiences in the early years of personal computing but also set a strong legacy for the way productivity and creativity support tools are still designed today.

Productivity tools such as Microsoft Word[1] and Microsoft Excel[2] are designed to streamline routine tasks. They can be explicitely structured, organizing the document in grids and layers, and surrounding it with menus, tabs, ribbons and feature panels that favor the discoverability of features, but also implicitely structured with a set of invisible rules. These structured designs usually rely on the use of keyboard shortcuts to speed up the workflow.

Creativity support tools interestingly aim to favor ideation while relying on similar highly structured interfaces tailored for efficiency. For instance, Adobe Photoshop[3] organizes tools in side panels for layers, colors, and brushes, to allow users to experiment with image manipulation while maintaining control over the content. Ableton Live's interface[4], divided into session and arrangement views, enables musicians to compose and modify tracks within an organized layout.

Unable to offer real flexible, unconstrained environments close to users' mental models [29], these tools rather embrace structure and sometimes propose a multitude of views and windows which are direct transpositions of physical tools' look (*e.g.*, color palettes, canvases, notebooks, paper, audio sequencers, synthesizers...).

Albeit powerful on desktop computers, tools designed for a given screen size and for mouse and keyboard modalities become suboptimal when brought to new devices with higher expressive potential without a proper reflection about the implications of different input modalities (*e.g.*, voice, gaze, pen, touch, mid-air gestures), output modalities (*e.g.*, display size and orientation, haptic feedback) and context of use (*e.g.*, individual or collaborative, stationary or mobile, indoor or outdoor). This can lead to serious usability issues, lower user engagement, and many missed opportunities [48, 99, 101].

1.3 ... and brought to Interactive Surfaces

Interactive surfaces like tablets are easy to carry and became increasingly popular for consuming content and taking notes. However, they are not so popular for editing content with productivity and creativity support tools, as those are derived directly from their desktop computer counterparts with

[1] https://www.microsoft.com/en-us/microsoft-365/word
[2] https://www.microsoft.com/en-us/microsoft-365/excel
[3] https://www.adobe.com/products/photoshop.html
[4] https://www.ableton.com/en/live

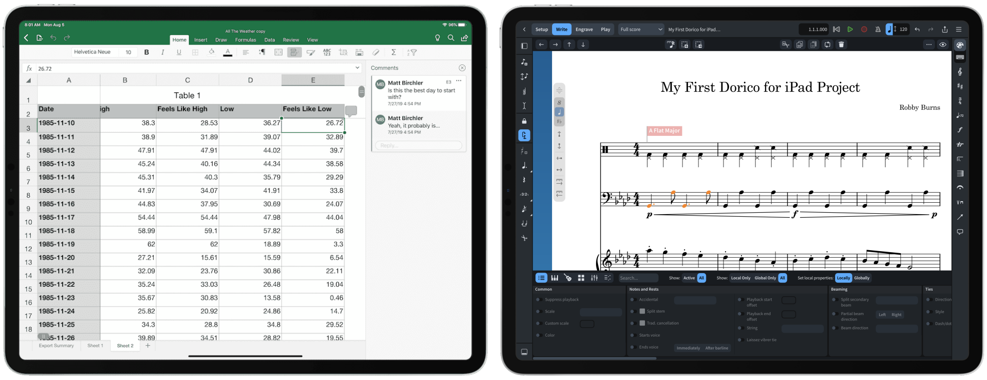
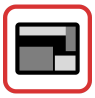
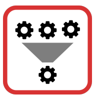


Figure 1.1: Microsoft Excel and Dorico [44] WIMP interfaces on iPad.

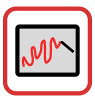
only minor adjustments. Instead of leveraging the unique affordances [61] of these devices, such as direct manipulation of content with fingers and expressiveness of the pen, tools like spreadsheet programs and music notation programs still impose a WIMP framework of indirect manipulations with persistent tabs, menus, and icons (see Figure 1.1) initially designed for pointing and clicking with the mouse. Computers with a capacitive screen like the Microsoft Surface come with additional challenges as they are half-computer and half-tablet: they must be usable both with a keyboard and a mouse, and with pen and fingers. With these medium-sized interactive surfaces, critical issues hindering content-editing applications happen on four different levels: content, tool, form factor and interaction.



Limited display area for content: Interactive surfaces with a smaller screen are easy to carry, but their screen real estate is precious, and pen, hand, and forearm are already known to cause significant occlusion [182]. A WIMP interface with elements surrounding the main content quickly takes an important portion of the display area and limits direct manipulation.



Limited number of features: A general band-aid solution to the previous issue is to design a simplified version of the tool with less features. A consequence of this is the reduced appeal toward users who still need a plethora of functionalities for their work and choose to stick to their traditional tools.



Fatigue from gestures: Interactive surfaces with a wider screen offer a better visibility of the content, but come with a proportionally wider motor space. Users performing long or repetitive gestures during an extended period of time can experience fatigue or even discomfort (*i.e.*, the Gorilla arm syndrome), especially when the surface is oriented vertically.

Friction from interactions: Regardless of screen size, an unclear design and unadapted structure leads to ambiguities and frustration when the expected outcome of an action does not match the actual result [158]. For instance, when pen and touch input are confused by the user (*e.g.*, gulf of execution [123]) or by the system (*e.g.*, palm rejection issues, accidental touch, lack of hover feedback).

Interactive surfaces encompass more than just tablets and touchscreen computers. Smaller devices like smartwatches and smartphones, tailored for mobile contexts, also belong to this category. However, due to their limited size and computing power, they are ill-suited for the rich functionalities that productivity or creativity require. On the other hand, larger devices such as tabletops, interactive whiteboards, and wall-sized displays are primarily designed for collaborative work. Both smaller and larger interactive surfaces have been extensively explored in the HCI literature, but there has been little work on medium-sized interactive surfaces which enable individual and in-depth work thanks to the digital pen.

1.4 Opportunities of the Digital Pen

Steve Jobs, famously hating the idea of using a stylus to interact on a screen, reportedly said in 1997 [82]:

> *"God gave us ten styluses, let's not invent another."*
> *- Steve Jobs, 1997*

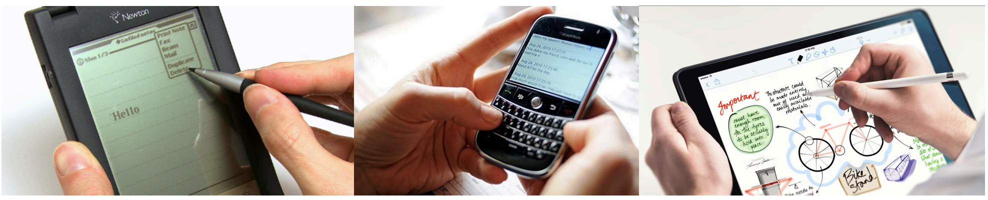


Figure 1.2: Left: Apple Newton MessagePad 110 (1994) [47]
Middle: Blackberry Bold 9000 (2008) [66]
Right: Apple iPad Pro (2015) [7]

His fight against styluses partly came from his resentment toward John Scully and his Apple Newton (see Figure 1.2-a), but was also reasonably rooted in their basicness: he did not see in them any real added value compared to fingers. It is true that touchscreens at the time did not support multi-touch and styluses' only purpose was to make up for the rudimentary capabilities of the resistive technology. They were pieces of plastic with a tip made of rubber, metal or conductive foam and essentially did what a nail could do. Unable to offer a fluid experience comparable to a real pen, their use remained mainly limited to help users maneuver the interface of Personal Digital Assistants. As a result of this, styluses were relegated in the background for a decade and device manufacturers focused their efforts on finger interaction with physical buttons and miniature keyboards (see Figure 1.2-b). However, in the early 2010s, capacitive touchscreens brought new opportunities. Substantially better both in resolution and brightness, they were made in various sizes and progressively became more accessible (see Figure 1.2-c). Evolving hand in hand with touchscreens, styluses were replaced by digital pens embedded with electronic components. They gained capabilities such as high precision and pressure sensitivity, approaching the familiar interactions we have with a real pen and offering features that fingers cannot. We may indeed have ten styluses, but we do not have ten digital pens. They enable:

Precise direct interaction: Digital pen allows for precise direct pointing and dragging, enabling users to manipulate small visual elements and reduce occlusion caused by touch interactions [144, 148, 181]. It also enables accurate input within graphically dense representations [199, 150].

Scribbling thoughts: Analog pen and paper allow users to capture fleeting ideas, annotate content, and take notes, which has cognitive benefits such as lowering working memory load and facilitating idea reformulation. This property is particularly important for sensemaking and data exploration tasks where numerous insights need to be tracked [148, 107, 114, 144, 159, 89, 131].

Cohabitation with touch: Systems supporting pen and touch offer the possibility to discriminate between the two types of input, allowing different roles to be assigned to each pointer and many more potential actions [110, 130]. When used together, pen and touch can unlock a third category of actions [76].

Sketching and drawing: Thanks to pencil and brush metaphors, digital pen is well-suited for quick sketching and creating more advanced drawings. Through expressive visualizations, digital ink can be used to enhance information presentation and aesthetics [194, 9, 146].

Handwriting: Pen input enables intuitive handwriting of text and symbols from any language, making it easy for a wide range of users to add content or annotations without the need for a physical keyboard or for an obstrusive on-screen keyboard.

Command marks: Using a recognizer, inked marks can be used to invoke commands [6, 11]. It improves *closeness of mapping* [15] in comparison with keyboard shortcuts. For instance, sketching a particular shape can create a specific type of visualization [93], and drawing lines between visualizations can activate coordination features [174].

Activable traces: Digital ink can be left as traces, which can be potentially activated later and multiple times to perform actions as both ActiveInk [148] and Musink [179] do.

Arbitrary strokes: Pen input facilitates the creation of arbitrary shapes, which is beneficial to accurately enclose a set of dense or sparse visual representations such as node-link diagrams, where precise and complex selections are required [148]. It can also be used to freely connect graphical elements [146].

In the scientific literature, these properties are often described as "natural". However, this term has been widely debated in theoretical studies (*e.g.*, [72, 103, 122, 124, 79]) because it can refer either to innate human abilities (*i.e.*, innateness) [97, 125] or to skills acquired through prior experience (*i.e.*, intuitiveness) [18, 186]. To avoid ambiguity, this thesis adopts the more specific terms **innate** and **intuitive** to describe the properties of the digital pen.

For example, the first three properties mentioned are predominantly innate rather than intuitive. Simple and direct input on a touchscreen is recognized as an innate ability [18]. Similarly, basic pen use as a means of self-expression and communication can emerge in children as young as two years old [98], and the preference for using the dominant hand for better motor control is deeply rooted in human evolution.

In contrast, the five other properties are more intuitive than innate. For instance, drawing, unlike scribbling, is a skill that must be learned before it becomes intuitive. Likewise, humans have not evolved to write [172]; handwriting and the use of language-related command marks are the results of deliberate training. Finally, tasks involving the selection or transformation of on-screen elements rely heavily on prior experience with user interfaces.

1.5 Research questions

Daniel Wigdor and Dennis Wilson describe in their book on "Natural User Interfaces" (NUI) [186] two goals for the design of intuitive systems: *"The first is that skilled use is obtainable very quickly. The second is that the interaction itself will feel enjoyable. In order to fulfill these promises, any NUI must be both efficient to learn and fun to use."* This is why they also recommend to *"start from scratch"*, as beginning with a successful GUI or Web interface and simply translating it into a NUI is likely to fail. Considering these principles, I will explore in this thesis the specific challenges and opportunities of designing pen-based interactions in two productivity and creativity contexts.

The main research question of my thesis is the following:

RQ0

How to design pen-based interactions to improve productivity and creativity on interactive surfaces?

To address this research question, we will operate under three assumptions:

1. Productivity depends on the efficiency a user can have to meet their goals (see Section 1.1).
2. Creativity depends on both efficiency and flexibility, as a user also needs to generate novelty (see Section 1.1).
3. The structure that is inherent to GUIs on desktop computers and beneficial when used with a mouse and keyboard is also the main breaking point when bringing content editing software to interactive surfaces. As detailed in Section 1.3, an inadequate structure leads to a bloated interface, less features, fatigue and friction, all of which negatively impact the users' workflow.

These considerations motivate the two more specific research questions of my thesis, targeting successively the productivity and creativity contexts:

RQ1

How can pen and structure be articulated in order to promote efficiency on interactive surfaces?

Spreadsheet programs such as Microsoft Excel and Google Sheets are amongst the best examples of productivity tools. Invented together with the emergence of personal computers, spreadsheet programs were among the first applications developed to transform tedious, manual tasks of managing, calculating, and analyzing data into fast, efficient processes. This is made possible thanks to their famous grid-based structured interface with the ability to handle very large datasets, making them irreplaceable in numerous fields like business, finance, research, and education. I will take them as a concrete use-case to answer RQ1.

> **RQ2**
>
> How can pen and structure be articulated in order to promote both efficiency and flexibility on interactive surfaces?

Symmetrically, music notation programs are good examples of creativity support tools. They provide a highly structured environment that enables composers to produce high-quality scores, but at the same time struggle to offer the flexibility needed to explore creative ideas and generate novelty. These tools, already challenging to use on desktop computers, cause additional problems when brought to interactive surfaces without rethinking the user's interaction with the structure. I will take them as a concrete use-case to answer RQ2.

1.6 Thesis outline

In this thesis, Chapter 2 reviews previous work on pen and touch interaction, followed by previous work on interaction with spreadsheets and music scores, with a particular focus on pen and touch contexts.

Chapter 3 explores a set of pen-based interaction techniques designed for efficiency in the use-case of spreadsheet programs. We first conducted an analysis of commercial spreadsheet programs and an elicitation study to understand what users can do and what they would like to do with spreadsheets on interactive surfaces. Informed by these, we designed interaction techniques that leverage the opportunities of the pen to mitigate friction and enable more operations by direct manipulation on and through the grid. We prototyped these interaction techniques and conducted a qualitative study with information workers who performed a variety of spreadsheet operations on their own data. Our observations show that using the pen to bypass the structure is a promising mean to promote efficiency with a productivity tool.

In Chapter 4, we explore a set of pen-based interaction techniques designed for both efficiency and flexibility in the use-case of music notation programs. We first conducted a series of 9 interviews with professional composers in order to take a step back and understand both their thought process and their work process with their current desktop tools. Informed by these, we elicited guidelines for the design of features which could promote both efficiency with frequent or complex operations and flexibility in regard to the exploration of ideas. Then, we acted on these guidelines by engaging in an iterative design process for interaction techniques that leverage the opportunities of the pen: two prototyping phases, a participatory design workshop, and a final series of interviews with 8 professional composers. Our observations show that on top of using the pen to bypass the structure for efficiency, using its properties to temporarily break the structure is a promising mean to promote flexibility with a creativity support tool.

In Chapter 5, I come back on the contributions of this thesis, the insights for designing pen-based interactions for productivity and creativity, and the future applications this thesis opens.

Chapter 2

Related Work

In this section, I review the literature that motivated and informed this thesis. I structure this literature along the following three axes: How pen and touch have been used on interactive surfaces, in particular with productive and creative applications (section 2.1); how information workers can interact with spreadsheets, with a focus on pen and touch interaction (section 2.2); and finally, how composers can interact with music scores, with a focus on their creative process and on pen and touch interaction (section 2.3).

2.1 Interacting with Pen and Touch

Early pen computing devices did not effectively support concurrent pen and touch input, and systems consequentially focused on using either pen [2, 90, 157] or touch input [42, 188, 193]. They also suffered from multiple problems including pointing accuracy, latency [3] and visual parallax [180]. Pointing accuracy is a particularly sensitive issue when editing documents, which are composed of many small elements. Hardware has improved significantly since then, and interaction techniques such as Pointing Lenses [137] let users select and manipulate interface elements with a high degree of motor precision.

Support for concurrent pen and touch input only came in the mid 2000s [197]. Devices that support both types of input need to differentiate between them and make the corresponding events available to application developers, creating one pointer for each contact point and tagging it accordingly (typically pen, eraser, and finger).

These devices also need to handle problems of unintended touch and palm rejection [4]. Here again significant progress has been made in recent years. Even if there is still room for improvement, users can most of the time safely rest the hand holding the pen on the interactive surface without triggering involuntary touch events. This is particularly important from a usability perspective, as being able to rest the hand causes less fatigue and enables a higher level of precision when selecting and manipulating on-screen elements [111].

An early study by Brandl *et al.* [22] identified several benefits to performing bimanual interactions with a combination of pen and touch, grounded in Guiard's kinematic chain model [67]. They discussed some foundational principles such as using the pen for precise input with the preferred hand and coarser actions (mode selection, parameter adjustments) with the non-preferred hand. They also empirically compared pen + touch to pen + pen and touch + touch, finding the pen + touch combination to perform better on a task that involved simultaneous inking and navigation. Informed by observations of people manipulating physical paper and notebooks, Hinckley *et al.* soon after introduced their *"pen+touch=new tools"* division of labor where *"the pen writes, touch manipulates, and the combination of pen + touch yields new tools"* [76]. In this paper, Hinckley *et al.* illustrate the general idea on an application for note-taking and scrapbooking. In fact, these two use cases are concrete examples that respectively target productivity and creativity, which are the two domains we focus on.

Investigations with Productive and Creative Applications

Pen and touch input has been investigated in a variety of applications aiming to support productivity and creativity. Many follow the same division of labor (*i.e.*, pen to write, touch to manipulate, and occasional combination of both to yield new tools): active reading [75], active diagramming [171] and annotating documents with systems such as RichReview [199] and SpaceInk [150]; handwritten text editing [174]; page-based document editing [110], working with maths [201]; and designing on a whiteboard [195].

Other research has investigated applications that depart from this general division of labor. In a laboratory experiment, Matulic *et al.* [111] found the pen to also be effective at performing some widget manipulation tasks, leading them to believe that the division of labor was *"not always clear-cut."* In their spreadsheet probe for thumb+pen interaction, Pfeuffer *et al.* make the pen select rather than write: *"the pen selects, touch manipulates"* [130] – acknowledging selection as one of the core transactions in the spreadsheet interaction model. In Bi-3D [129], the pen is used to ink shapes in 3D while the non-preferred hand navigates. The pen can also be used to select and manipulate existing vector shapes, for instance to edit node-link diagrams [53]. Graphies [146] – an expressive network visualization authoring environment – uses the pen primarily for selection and for parameter adjustments. In Neat [54] – a set of pen and touch techniques for vector graphics layout – the pen is used not only to sketch shapes but to draw alignment guides as well. Overall, we observe that beyond writing, *selection* comes a close second as a role for the pen. It is indeed a particularly effective precision tool for delineating arbitrary regions, which will prove particularly useful for interactions with spreadsheets and music scores.

Beyond node-link diagram editing, pen and touch has been investigated in a large number of data visualization systems, which are more extensively covered by Lee *et al.*'s comprehensive survey [94]. Pen and touch visualization systems require users to manipulate many small interface elements and to perform elaborate selections. Among the first systems, SketchVis [25] only supported single-pen input, but it introduced interesting ideas. For instance, aiming to keep the gesture set simple, users could filter items in a chart by stroking out their category label in the legend or could apply an operator to the data by circling it. Following up on observations from a Wizard-of-Oz study [183] in which participants used pen and touch to visually explore data, Lee *et al.* designed SketchInsight [93]. They observed that participants had a fairly clear idea about when to use pen and when to use touch, sometimes intentionally stowing the pen in the preferred hand's palm to interact with touch. In SketchInsight, the pen is used to draw charts and make annotations. Touch selects chart elements, and manipulates them. Touch can also be used to get details-on-demand from such charts, as Subramonyam & Adar have shown with SmartCues [170]. In other data analytics systems such as PanoramicData [202], precise selections are performed with the pen, which is also used to connect components of the analysis workflow by drawing links between them [40].

Pen and touch control different pointer types, a general trend in many systems is to use them for different types of actions, helping disambiguate input without requiring an explicit mode switch. Observing here again that *selection* is a fundamental operation, Sadana & Stasko [155] let users perform elaborate selections using both hands, the non-preferred hand essentially serving as a basic mode switch similar to the spring-loaded modes from [130]. InChorus [165] integrates many of these ideas together, letting users interact with visualizations on tablets using a pen in their preferred hand, touch in their non-preferred hand, as well as speech input. Again, users perform selections primarily with the pen. ActiveInk [148] explores a fairly different interaction model, where ink left in the workspace by the pen can serve multiple purposes: drawing and annotating, bookmarking, as well as selecting items and invoking commands on them. Here again, selections are performed using the pen, observing that one *"key affordance of digital pens […] is the accuracy for direct pointing and dragging, providing the ability to compose precise selections by sketching complex shapes […] while reducing the occlusion caused by touch"* [148]. Prior work discussed so far has involved digital pens in combination with direct touch input. In the last few years, research projects have looked into expanding input capabilities, for instance with gaze [127, 128], speech [165], pen grip and hand posture [112, 28, 109] or even tablet grip [204] in search of ever more innate, intuitive, and rich interactions.

2.2 Interacting with Spreadsheets

Spreadsheets have received significant attention from the HCI research community. Research works include studies about how information workers use regular spreadsheet programs (see, *e.g.*, [10, 33]); solutions to effectively deal with errors and hidden dependencies (see, *e.g.*, [81, 156, 196]); algorithms that help automatically transform data [70, 73, 84]; and interaction techniques for the direct manipulation of those data. In this review of the state of the art, we focus on the latter only.

Early research work about user interfaces for tabular data manipulation investigated ways to support users in making sense of large tables. The Table Lens [138] adopts a focus+context strategy [37] to embed visual representations of the data into the grid-based representation of the table, letting users sort and filter, get detailed symbolic representations of rows and columns of particular interest and identify values that characterize the distribution [132]. Interaction with the table involves keyboard shortcuts and a pointer, but includes basic flick gestures performed with the mouse as well. FOCUS [164] also uses a focus+context strategy to show large tables and embed visual representations of some of the data, but rather focuses on supporting comparison tasks and exploring large datasets by restricting the number of items to show.

While the above tools are mostly about *visualizing* tabular data, other tools are rather aimed at interactively *manipulating* those data. Data wrangling tools such as Potter's Wheel [136] and Wrangler [86] let people clean, restructure and edit data, inferring transformations from user interactions. While very powerful and useful, these tools rely on a classic WIMP-oriented spreadsheet-like interface.

Pen and Touch for Spreadsheets

With the advent of interactive surfaces, researchers have started investigating how to use the pen and touch modalities for spreadsheet manipulation. Tableur [203] aims to support the quick creation of relatively simple spreadsheets away from the office workstation. The pen is used to sketch the table by hand on a blank canvas and then input cell values. Gestures trigger commands such as the recognition of the digital ink, the erasure of content, and

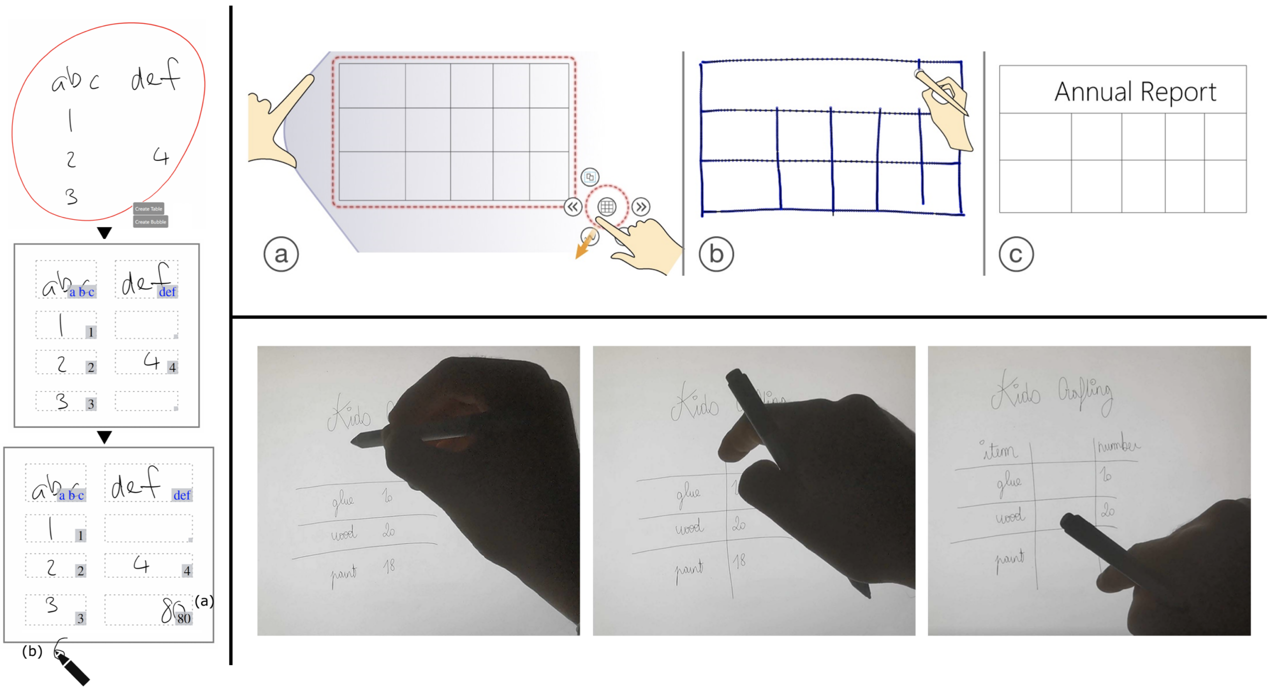
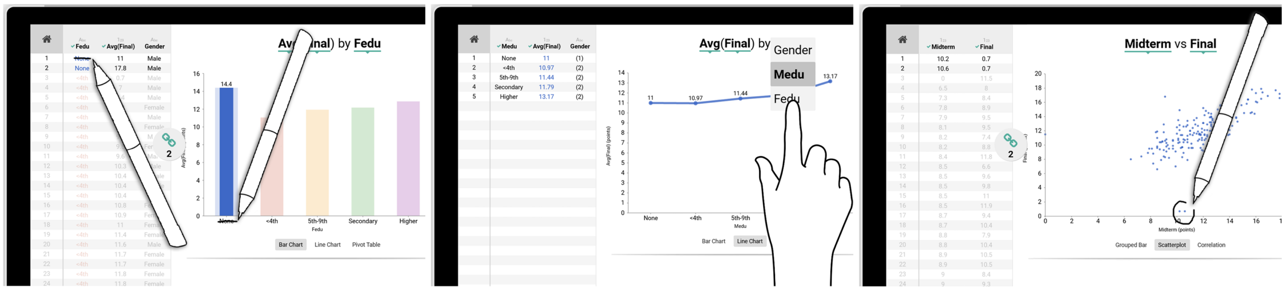


Figure 2.1: Left: **Tableur** [203] lets users sketch values by hand on a blank canvas, segment them into a table which can then be edited.
Top right: **WritLarge** [195] lets users seamlessly switch between a formatted table (a) and a hand-drawn grid to edit the strokes (b). Handwriting on a formatted table is immediately recognized as text (c).
Bottom right: **Style Blink** [149] lets users create simple tables by sketching and interact with the strokes to move content.



(a) (b) (c)

Figure 2.2: TouchPivot [85]. (a) Users can filter out a category by striking out the category on the table or visualizations. (b) Users can change the columns in a pivot transformation by tapping on the names in a chart title and selecting other columns. (c) Users can draw a lasso on a scatterplot to see details on the points contained in the lasso. Similarly, they can draw a circle on the table to locate the selected records on the scatterplot.

the propagation of a formula over a range of cells. It focuses on small, informal spreadsheets that can reasonably be drawn by hand. The WritLarge [195] early-stage design system for electronic whiteboards also enables users to create simple tables by sketching among many other things. The digital ink that represents the table can be selected with the non-preferred hand and semantically *"elevated"* thanks to a simple heuristic that recognizes the hand-drawn grid and turns it into an actual table structure. A very interesting property of this approach is that users can go back and forth between ink and formatted table, editing strokes at the ink level to merge some cells, for instance. But again, this approach primarily supports the creation of small tables drawn by hand. Further exploring the idea of creating and manipulating semi-structured information with digital ink, Style Blink [149] demonstrates how the reification of ink styles helps users create simple tables and change their appearance and layout. These systems are illustrated in Figure 2.1.

While the above works essentially focus on sketching simple spreadsheets on a blank 2D canvas, other related work has investigated the use of pen and touch to manipulate data in conventional spreadsheets programs. As part of their solution to interactively repair tables extracted from documents on mobile devices, Hoffswell & Liu [77] describe a small set of gestures that users can perform to insert and delete cells, as well as split or merge cells that were incorrectly recognized by the automatic extraction process. Although not a spreadsheet program but rather a tool for visual data exploration, TouchPivot [85] (see Figure 2.2) features a table view that users can interact with to perform some transformations on the data. The focus is specifically on filtering and pivoting data using simple pen and touch interactions. Pfeuffer *et al.*'s investigation of thumb+pen interaction on handheld tablets [130] includes a spreadsheet application probe that enables users to perform a set of cell manipulations using the pen primarily as a selection tool, while the non-preferred hand complements that input by setting spring-loaded modes and adjusting parameters.

Finally, other modalities beyond pen and touch have been investigated to interact with spreadsheets. Gesslein *et al.* [60] use a virtual reality headset to extend the limited display space of a tablet in a mobile context of use. The additional display capacity is used to show multiple worksheets, to extend the current worksheet, and to superimpose additional information such as dependencies between cells using the third dimension. Many interactions are performed with the pen, though some actions are performed with mid-air and touch gestures. Takayama *et al.* [173] run an elicitation study to design a set of contactless gestures for mid-air spreadsheet manipulation in front of a regular desktop monitor. Finally, Perelman *et al.* [126] use a smartphone to perform grid-level selections on tablets, observing that touch interaction alone is limited to a few gestures, many of which are reserved for navigation actions.

2.3 Interacting with Music Scores

In Western music, paper scores have long served as the primary medium through which composers communicate their musical pieces to performers (*i.e.*, instrumentalists, singers). They not only act as a blueprint for the music but also guide performers in interpreting and realizing the composer's intentions.

However, for several decades, composers and performers are increasingly embracing technological advancements to augment their experiences with scores. Digital tools and platforms now allow for interactive scores that streamline music composition, facilitate collaboration, and provide performers with new ways to engage with and interpret music. Listing those tools and platforms goes beyond the scope of this manuscript, we rather rely on Masu *et al.*'s taxonomy [108] to delineate the category we are interested in. Based on articles published in the NIME conference proceedings[1], Masu *et al.* actually identify five uses of scores: (1) Scores as *instructions* – giving instructions in real-time to a performer who is playing or learning how to play an instrument; (2) Scores as an *interface to play a Digital Musical Instrument* – functioning as an input for interactive systems; (3) Scores as *synchronization* – listening to the performer(s) and acting as a coordination tool; (4) Scores as a *recording* – capturing a performance without affecting it; and (5) Scores

[1]New Interfaces for Musical Expression, https://www.nime.org/

creation – supporting the creative process of composers. In this section, we consider only the latter and focus on scores both as a tool and as a result of compositional activity.

In order to better understand how composers interact with their music scores, it is essential to step back and consider the creative process of music composition as a whole. Bennett [14] conducted one of the first qualitative studies about the musical creation process, interviewing eight professional composers. Each composer has their own process, but high-level activities can be identified across composers. The discovery of a *"germinal idea"* often comes first. And then the drafting of this idea, elaborating upon and refining it – what is identified as the *writing* and *rewriting* phases in another study by Roels [145]. Rosselli del Turco and Dalsgaard [151] provide a good overview of additional studies that aimed at better understanding composers' creative process [50, 184], and contribute their own study about how music artists capture and manage their ideas. They draw a parallel with personal information management tools, and discuss recommendations for the design of tools aimed at supporting idea management in music.

Key takeaways from the above studies of particular interest here can be summarized as follows:

- even if the creative process can be characterized in terms of overall strategy [50] and high-level activities [14, 145, 184], it varies significantly from one composer to another and across music genres [50];
- the process is highly iterative [38], composers writing and rewriting [145, 16], spending much time elaborating and refining their draft [14].

Coughlan and Johnson [39] discuss the tension between the desire for flexibility of composers and the constraints imposed by software tools that stems from interaction design choices made by the developers of those tools, and from the need for these tools to be able to interpret user input. While they made this observation about composition tools that do not use staff notation as the primary means to represent music, this tension clearly exists in music score editing software as well. Many of the programs used by composers (Finale[2], MuseScore[3], Sibelius[4]) impose significant constraints on how composers can input and modify the notation, impeding the creative process.

In their "Design Workbench for Interactive Music Systems", Malloch *et al.* [105] point to three models from the HCI literature that are particularly relevant for musical interaction: Jens Rasmussen's *Human Information Processing* [104, 139], Beaudouin-Lafon's *Instrumental Interaction* [12], and

[2] https://www.finalemusic.com/
[3] https://musescore.org/en
[4] https://www.avid.com/sibelius

Mackay's *Co-adaptation* [102]. They also present tools that have been explicitely designed to support these models [57, 58] and suggest eight design guidelines[5] for the design of interfaces for musical expression in advanced musical contexts. The first six guidelines rather apply to interfaces for musical performance, while the last two apply to our context of score creation: (1) composers need flexible tools that support their personal strategies and conceptual representations, and (2) tools that can support the various representations involved during the different phases of the creative process, each with specific advantages for the task at hand (*e.g.*, a canvas is great for sketching ideas, but a system with staves is better suited for concretizing them).

Some previous work has explored such tools. Freeman [52] investigated with Graph Theory the use of a Web-based interface for crowd-sourced composition. Users navigate through audio musical fragments and create their own preferred path. Each day, an algorithm assembles the most popular path segments and automatically turns them into a score. Manesh & Egozy [106] explored with Exquisite Score a different type of collaborative Web-based interface. Through a MIDI-sequencer interface, Exquisite Score allows each composer to contribute a section to a piece of music, only showing them the very end of the preceding section, written by another composer. Focusing on systems meant for individual usage, Laurson & Kuuskankare [92] presented the concept of "macro-notes" to enrich the scripting syntax of music notation with auxiliary note information, such as playing techniques and gestures, otherwise impossible to notate and listen with traditional tools. Similarly, Resch [141] describes note~ for Max [135], a library for the Max/MSP environment that empowers composers with precise edits of note parameters, such as arbitrary pitch and duration, relying on the ability to switch between score view and piano-roll view. In a follow-up article, Resch & Bilbao [142] detail the changes made on note~ since its first iteration.

Pen and Touch for Music Scores

Leroy *et al.* [95] were among the first to develop an interactive system using an early form of optical music recognition [27] to let composers use handwriting to input music symbols on staves. From the very beginning, Leroy *et al.* explicitly state that this is *"a system for composers, not for engravers,"* observing that *"composers are skilled in handwritten notation and editing and need to sketch out musical ideas rapidly while retaining the ability to make alterations at both local and global levels"* [95]. Presto [5, 119] is another early pen-operated system for music score editing. The system was designed to facilitate the input of music symbols on staves, this time using a set of simple, predefined gestures rather

[5]The first four guidelines were introduced by Hunt and Wanderley in [80].

Symbol/effect	Gesture	Context
default note		Dot on pitch
default note, stem up		Start on pitch, draw up then down
default note, stem up		Start on pitch, draw up then down
double default duration		Double dot on pitch
quaver rest		On staff
crotchet rest		On staff
minim rest		On staff
semibreve rest		On staff
barline		Over staff
double bar-line		Next to single bar-line
♯ sharp, 𝄪 double sharp		Start on note-head
♭ flat, 𝄫 double flat		Start on note-head
Change stem up		Start on note-head, draw up then down
Change stem down		Start on note-head, draw down then up

Figure 2.3: Presto2. An extract from the table of gestures presented in [119].

than handwriting recognition (see Figure 2.3). Several other manipulations were possible, such as moving elements, transposing or adding ornaments, but accessible via menus only.

Following early observations about contemporary composers' use of paper and computer by Letondal & Mackay [96], Tsandilas *et al.* developed Mus*ink* [179]. Mus*ink* is a computer-based composition environment that integrates sketching, gesturing and end-user programming in the same workflow thanks to interactive paper [74] (see Figure 2.4) with *Anoto* technology. It allows composers to define their own vocabulary of annotations on musical scores, which can then be interpreted as functions. Garcia and colleagues further explored this line of research, designing creativity support tools that combine paper and computer programming to support contemporary composers' creative process such as InkSplorer [56], Paper Substrates [57], PaperTonnetz [55], PaperComposer [58] (see Figure 2.5). They later developed Polyphony [59], an interface supporting the study of these complex processes. These works have primarily focused on contemporary composers, who may use the staff notation in certain aspects of their work but also heavily rely on music programming tools such as Max and OpenMusic [23], as well as on notations that they design themselves.

On the commercial side, several music composition programs designed for the desktop can operate on interactive surfaces that support digital pen input, including Sibelius, Dorico, Flat, and Notion (see Figure Figure 1.1). A few programs have also been designed specifically to run on interactive surfaces, such as StaffPad [166] and Symphony Pro [134]. Both categories essentially rely on the traditional WIMP model of desktop programs, making very few

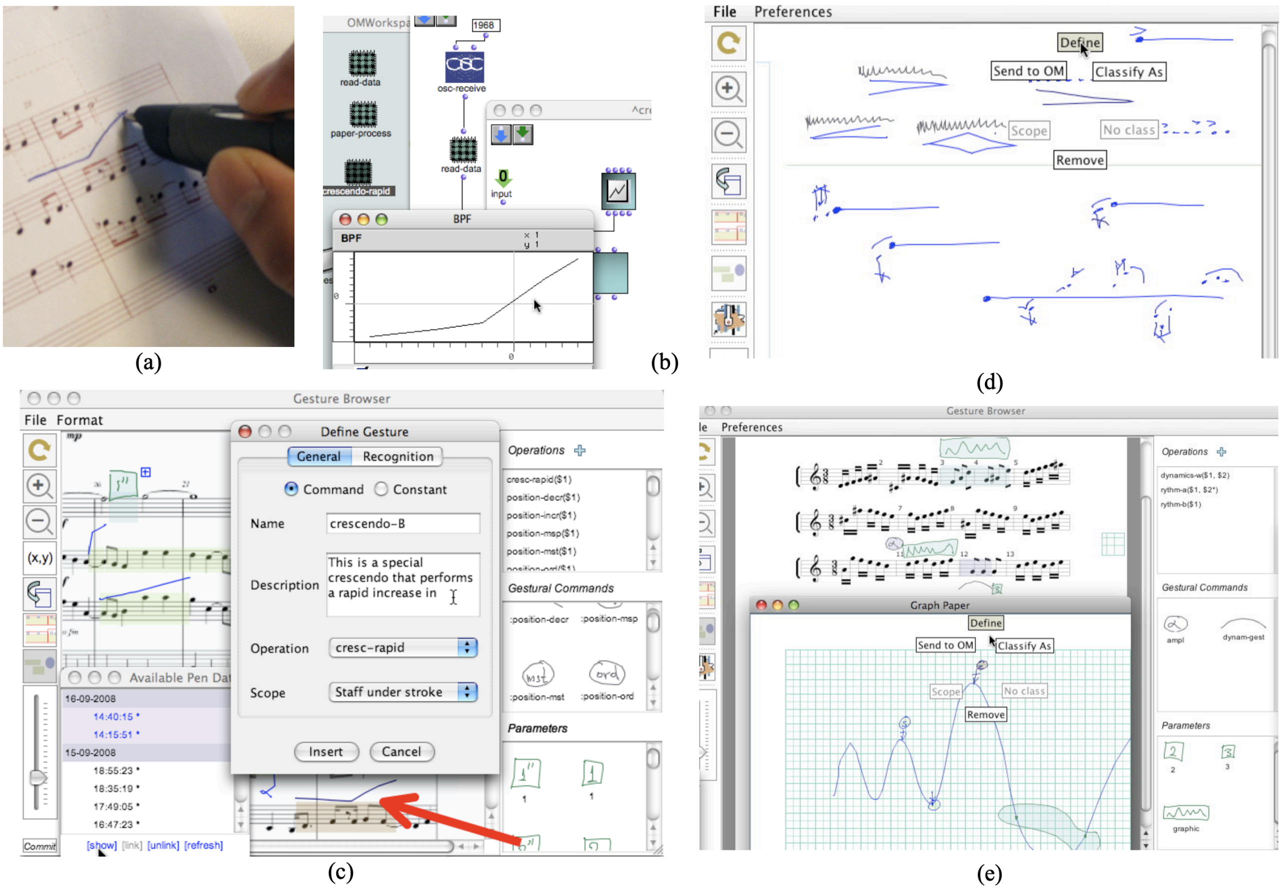


Figure 2.4: Mus*ink* [179]. On the left, a scenario: (a) Drawing on paper: expressing a new type of crescendo (b) OpenMusic: defining the crescendo's vibration pattern (c) Mus*ink* Gesture Browser: defining the crescendo class.
On the right, (d) a score created by a composer during an interview is shown on the Gesture Browser (e) a representation of Mus*ink* gestures linked with detailed graphs, supporting functions over curves.

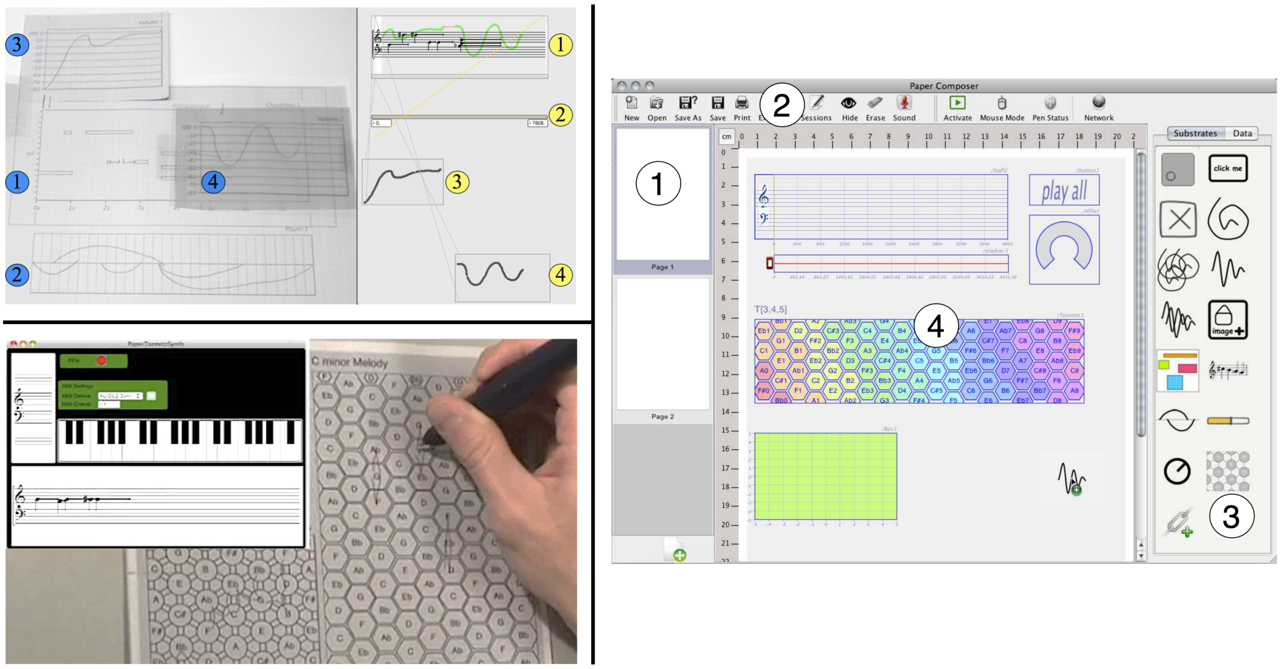


Figure 2.5: At the top left, **Paper Substrates** [57] used to edit a musical sequence on a printed interface with physical data (blue) and on the Max/MSP interface with digital data (yellow).
At the bottom left, **PaperTonnetz** [55] used to create and listen to a musical sequence drawn on a two-dimensional heptatonic layout printed on interactive paper.
On the right, **PaperComposer** [56], a graphical interface builder that allows users to create a personalized interactive paper interface that they can connect to their own computer-based musical data.

adjustments to it. Pen and fingers are seen as generic pointing devices that can be used interchangeably, and composition programs treat them as mere alternatives to mouse or trackpad. They do not fully leverage their specific affordances and expressive power.

In this context, the pen is little more than a mouse with a single button, making numerous mode switches necessary. It still has interesting properties though, most notably when inking on a canvas as it better affords handwritten input. The few music composition programs designed specifically for interactive surfaces thus support handwritten input of common music symbols. This capability is highly valuable as it makes the experience closer to writing music on paper. But multiple constraints still impede the creative process. Two constraints relate to the ink recognition process. First, symbols need to be inked in a predefined order to ensure that the recognition engine will be able to interpret them correctly, imposing a particular way of writing on composers. Second, recognition is implemented as a greedy process, beautifying handwritten symbols as soon as a measure is complete and enforcing syntactic well-formedness rules on beautified symbols systematically. For instance, a measure must be filled before proceeding to the next one. The user experience is thus closer to writing music on paper than what desktop programs afford, but the flexibility associated with writing music freely on paper is lost. Those constraints limit composers' freedom to experiment and jot ideas down [118]. Composers need to follow certain rules when inking the notation, and once input, the notation remains difficult to edit.

Chapter 3

Bypassing the Structure for Productivity

This chapter is based on a full paper written in collaboration with Caroline Appert and Emmanuel Pietriga, "Spreadsheets on Interactive Surfaces: Breaking through the Grid with the Pen" [30], published in TOCHI: ACM Transactions on Computer-Human Interaction. Supplementary material and video for this project are available at https://ilda.saclay.inria.fr/pw2/supplemental_material.

Spreadsheet programs for interactive surfaces have limited manipulations capabilities and are often frustrating to use, resulting in an inefficent workflow. One key reason is that the spreadsheet grid creates a layer that intercepts most user input events, making it difficult to reach the cell values that lie underneath. In this chapter, we conduct an analysis of commercial spreadsheet programs and an elicitation study to understand what users can do and what they would like to do with spreadsheets on interactive surfaces. Informed by these, we design interaction techniques that leverage the precision of the pen to mitigate friction between the different layers. These enable more operations by direct manipulation on and through the grid, targeting not only cells and groups of cells, but values and substrings within and across cells as well. Finally, we prototype these interaction techniques and conduct a qualitative study with information workers who perform a variety of spreadsheet operations on their own data.

3.1 Motivation

In his reflections about the use of spreadsheets in organizational life, Paul Dourish observes that *"a spreadsheet starts not blank but empty"* [45]. This intriguing statement captures a key characteristic of spreadsheet programs. The *grid* effectively creates a layer above the values, which plays a central

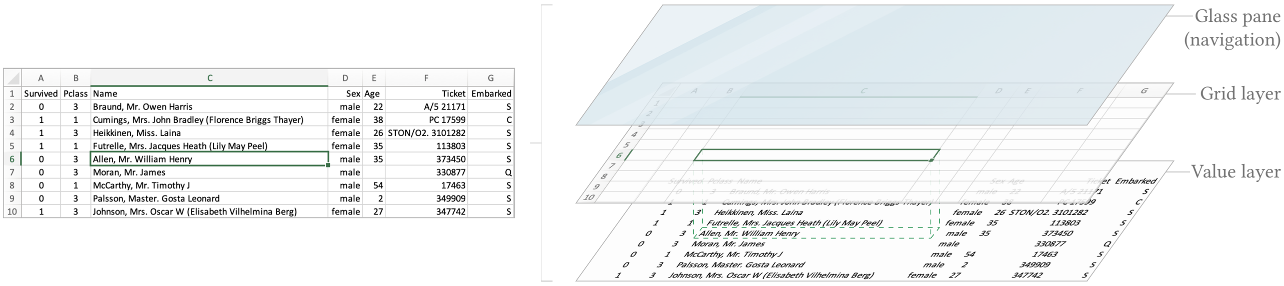


Figure 3.1: The three layers of a spreadsheet program's user interface on an interactive surface.

role in enabling many of the direct manipulations that users perform on the spreadsheet's columns, rows and cells. But because it covers the entire workspace, this grid layer captures most input events, often complicating straightforward tasks like selecting a substring within a cell. Additionally, spreadsheet programs designed for interactive surfaces introduce yet another layer that captures basic touch actions for the purpose of navigation.

These three layers, illustrated in Figure 3.1, create tension between the different types of interactions: *grid-level* interactions (cell, row and column selection & manipulation); *value-level* interactions (text, number and formula editing); and *navigation* interactions (panning and zooming the worksheet). This tension between layers often breaks the direct manipulation paradigm, where users expect to effortlessly select an element and manipulate it. This tension exists even when performing the most elementary action: *selection*. While it is easy to select a single cell, selecting only part of a text or number inside a cell is tedious, requiring multiple actions to traverse the grid and adjust the selection. Additional problems occur when selecting sets of elements. Selecting elements such as, *e.g.*, rows or columns is possible but interferes with navigation actions when all elements are not visible simultaneously. And while spreadsheets are designed to manipulate sets of cells, it is never possible to select only a part of the value across multiple cells at once. Such manipulations are useful though, for instance to select a specific suffix such as the country code in a list of cities to remove them all, or to separate surname from firstname in a list of people with a single manipulation.

This chapter introduces a set of novel direct manipulation techniques that enable seamless interaction at both grid and value levels, within and across cells, as illustrated in Figure 3.19. With these techniques, users can achieve diverse operations with a few pen marks and touch actions. Building upon previous work on pen + touch [76, 130], we implement a clear division of labor between input modalities, dedicating the pen to selection, touch to manipulation, and multi-touch to navigation. A key element of our approach is to use the pen not only as a tool to select grid-level elements, but as a tool to *break* *through the grid* as well, enabling precise, seamless value-level selections without mode switching. We further leverage the pen's unique precision to design small pen-operated widgets that users can invoke to jump to distant locations in the spreadsheet or to perform advanced, non-contiguous selections of tabular data rows and columns.

We first examine a representative sample of commercial spreadsheet programs over a range of operating systems and pen and touch hardware: handheld tablets, slate PCs, large digital drawing boards. We highlight inconsistencies among them, as well as the main sources of friction between layers. We then conduct an elicitation study to gather empirical data about users' expectations when using pen and touch to interact with spreadsheets. Informed by both our analysis of commercial spreadsheet programs and our elicitation study, we design a set of interactions that enable seamless access to all layers, leveraging the expressive power of pen and touch input to enable the *effortless selection of a variety of elements in spreadsheets*. These elements include individual cells, groups of cells, as well as values and substrings within cells or spanning multiple cells. We implement this set of interactions in a prototype that we use to conduct a semi-structured qualitative study with six information workers performing a variety of spreadsheet operations on their own data. We discuss how participants used the interactions, and how their feedback helped us improve on them. We conclude with opportunities to explore as future work.

3.2 What Users *Can* Do with Commercial Spreadsheet Programs

Microsoft Excel, Apple Numbers and Google Sheets – to name well-known commercial spreadsheet programs – are all available on tablet computers. Their cumulative download counts amount to billions. These spreadsheet programs typically have less features than their desktop counterparts, but their UI design and interaction model is actually very similar, which is one of the causes of friction between the grid, value and navigation layers. Table 3.1 summarizes our systematic analysis of input-to-action mappings over multiple *configurations*: different spreadsheet programs (Microsoft Excel, Apple Numbers, Google Sheets) and operating systems (Microsoft Windows, Apple iPadOS, Apple macOS), running on the three representative pen and touch devices illustrated in Figure 3.2.[1] The effect of each input action is categorized into: grid-level selections (GS_*) and manipulations (GM_*); value-level selec-

[1]The raw table of all input events and actions triggered, per configuration, is provided at https://ilda.saclay.inria.fr/pw2/supplemental_material/, together with information about spreadsheet program version and hardware configuration.

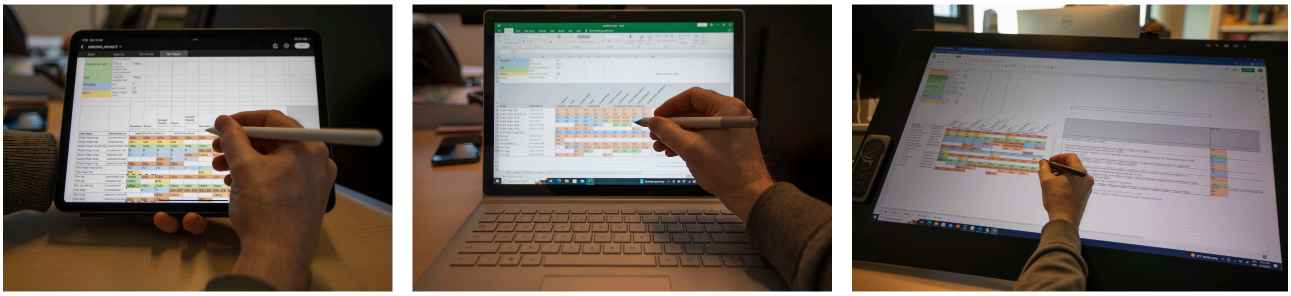


Figure 3.2: Sample configurations used in the analysis of commercial spreadsheet programs: *Numbers* running on an Apple iPad Pro; *Excel* running on a Microsoft Surface Book; *Google Sheets* running on a Wacom Cintiq Pro.

tions (VS$_*$); panning & zooming the worksheet (P, PZ); and invoking contextual widgets (MN). Looking at this table, we can make several observations.

Lack of consistency One first observation is the inconsistency [120] between input-action mappings across configurations. Looking at individual rows, color variations reveal significant differences between them. There are variations among hardware configurations for a given spreadsheet program, and among programs for the same hardware configuration. These variations are symptomatic of the friction between the three interaction layers (grid, value, navigation), different spreadsheet program UI design teams addressing the problem in different ways.

Friction between grid and navigation layers A typical example of friction that results in a variety of mappings across configurations is the case of `drag` input events. Starting with the case of single-finger drags performed on already-selected cells, Table 3.1 lists four different actions (l.05),[2] illustrated in Figure 3.3. Considering all drag input events (l.04-07 + l.13-15), the same table shows that regardless of the modality (finger or pen), these are sometimes mapped to grid-level selection actions (GS$_*$, GM$_*$) and sometimes to navigation actions (P). There is broad agreement regarding two-finger drag/pinch input events however: these are systematically mapped to navigation (l.08).

Friction between grid and value layers Table 3.1 also shows that selections tend to be cumbersome no matter the level considered (grid or value). For instance, selecting a range of cells (a grid-level action) typically involves two steps: first, a cell is selected (GS$_n$) with a tap (l.01+l.10), followed by a drag on a small, difficult-to-acquire element such as the selection border or lower-right corner handle (l.06-07 + l.14-15) to adjust the selection (GS$_a$). Value-level

[2] Notation **l.nn** refers to line numbers in Table 3.1.

	User input	Performed on	Apple iPad Pro + Pencil 2: Numbers (iPadOS)	Apple iPad Pro + Pencil 2: Excel (iPadOS)	Apple iPad Pro + Pencil 2: Google Sheets (iPadOS / Safari)	MS Surface Book: Excel (Windows)	MS Surface Book: Google Sheets (Windows / Chrome)	Desktop computer + Wacom Cintiq Pro: Numbers (macOS)	Desktop computer + Wacom Cintiq Pro: Excel (macOS)	Desktop computer + Wacom Cintiq Pro: Google Sheets (macOS / Chrome)	Desktop computer + Wacom Cintiq Pro: Excel (Windows)	Desktop computer + Wacom Cintiq Pro: Google Sheets (Windows / Chrome)
l.01	Single-finger tap	unselected cell	GS_n	GS_n	GS_n	GS_n	GS_n	GS_n	GS_n	GS_n	GS_n	GS_n
l.02	Single-finger tap	selected cell	MN	MN	MN	MN			MN			
l.03	Single-finger double tap	(un)selected cell	VS_{lc}	VS_{lc}	VS_{lc}	VS_{lc}	VS_{lc}	VS_{nc}	VS_{nc}	VS_{lc}	VS_{lc}	VS_{lc}
l.04	Single-finger drag	unselected cell	P	P	P	P	P	GS_n,GS_a	GS_n,GS_a	GS_n,GS_a	P	P
l.05	Single-finger drag	selected cell	P	P	P	P	GS_a	GM_{cp}	GS_n,GS_a	GS_n,GS_a	P	GS_a
l.06	Single-finger drag	selection border			GS_a	P	P		GM_{cp}	GM_{cp}		
l.07	Single-finger drag	selection handle	GS_a	GS_a	GS_a	GS_a					GS_a (*)	
l.08	Two-finger drag/pinch		PZ	PZ	PZ	PZ	P	P	P	P	PZ	P
l.09	Two-finger tap		MN					MN				MN
l.10	Pen tap	unselected cell	VS_{nc}	GS_n	GS_n	GS_n	GS_n	GS_n	GS_n	GS_n	GS_n	GS_n
l.11	Pen tap	selected cell	VS_{nc}		MN							
l.12	Pen double tap	(un)selected cell	VS_{ac}	VS_{lc}	VS_{lc}	VS_{nc}	VS_{lc}	VS_{nc}	VS_{nc}	VS_{lc}	VS_{nc}	VS_{lc}
l.13	Pen drag	(un)selected cell		P	P	GS_n,GS_a	GS_n,GS_a	GS_n,GS_a	GS_n,GS_a	GS_n,GS_a	GS_n,GS_a	GS_n,GS_a
l.14	Pen drag	selection border			GS_a	GM_{cp}			GM_{cp}	GM_{cp}	GM_{cp}	
l.15	Pen drag	selection handle	GS_a	GS_a	GS_a			GS_a			(*)	

(*) The selection rectangle features handles only if it was created using finger touch (tap for single cell select).
⬚ Freeform inking available by switching to a dedicated annotation mode.

GS_n	grid-level selection *(new cell selection)*	VS_{ac}	value-level selection *(value editing: select all characters)*	P	pan sheet
GS_a	grid-level selection *(adjust cell selection)*	VS_{lc}	value-level selection *(value editing: position caret after last character)*	PZ	pan & zoom sheet
GM_{cp}	grid-level manipulation *(cut & paste selected cells)*	VS_{nc}	value-level selection *(value editing: position caret after nearest character)*	MN	invoke context menu

Table 3.1: Summary of user actions and their effect, by spreadsheet program, by operating system, by display surface hardware. Actions are grouped in four categories, which are visually encoded with redundancy to help identify them based on the action code's first letter and background color: grid-level selection GS_* and manipulation GM_*; value-level selection VS_*; P, PZ panning & zooming the worksheet; MN invoking contextual widgets. Empty cells indicate that this input has no effect.

selections are cumbersome as well, requiring a double tap to get through the grid layer and interact with values on the layer below, one at a time. In addition, while this double tap is mapped consistently to a value-editing mode-switch across configurations, the actual selection state resulting from this switch varies (l.03, l.12). As shown in Figure 3.4, depending on the configuration considered, a pen double tap can either automatically select the entire string, position the text insertion caret at the end of the string, or position it between the two characters nearest to the double-tap location. In all cases, selecting a specific substring or moving the caret to another position requires additional input actions. On the desktop, this is facilitated by the keyboard and the mouse's multiple buttons. But none of these is available – or easily accessed when available – on interactive surfaces.

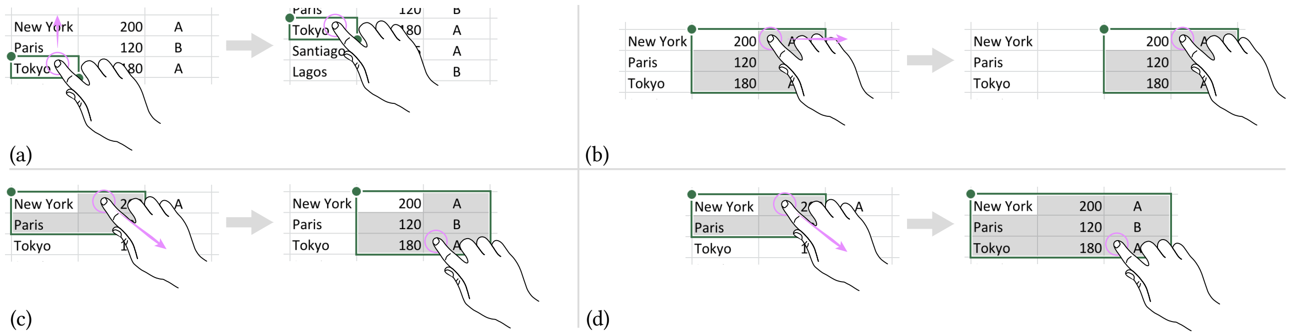


Figure 3.3: Four different actions triggered when dragging with a single finger on an already-selected cell: (a) pan the sheet P; (b) cut & paste the content of the cells from the current selection GM_{cp}; (c) reset the selection and then adjust it GS_n,GS_a; (d) adjust the current selection without resetting it GS_a.

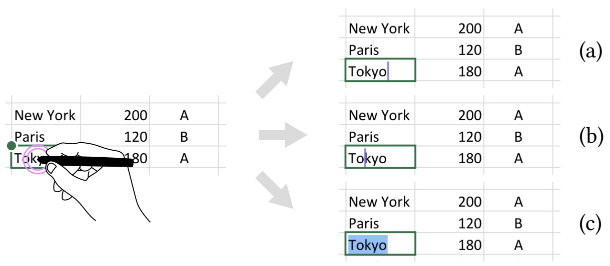


Figure 3.4: Three different behaviors when performing a pen double-tap between characters `o` and `k` in cell `Tokyo`: enter value-edit mode, and either (a) position the caret at the end of the string (VS_{lc}); (b) position the caret in the string, precisely between those two characters (VS_{nc}); or (c) select the whole string (VS_{ac}).

In summary, spreadsheet programs for interactive surfaces lack in clarity when it comes to disambiguating grid-level interactions (cell selection and manipulation), value-level interactions (string editing), and navigation interactions (panning and zooming the worksheet). There are multiple inconsistencies across programs, and many actions – including some very basic ones – are cumbersome. The only consensual design choice we observe is that *two-finger interactions are consistently dedicated to the navigation layer* ($\mathbf{Obs_1}$). Apart from that, there is no clear model for the fundamental action of selecting elements, which is key to any direct manipulation technique. We argue that the difficulty in selection is mainly due to spreadsheet programs following *"a legacy of designs that have treated pen or touch interchangeably"* [76], missing opportunities to leverage the specific capabilities of pen input.

3.3 What Users *Could* Do with Spreadsheet Programs

The above analysis gives us an overview of what actions users *can* do with current commercial spreadsheet programs operated with pen and touch. We now start considering what actions users *could* do with those input modalities. We first report on what strategy they would adopt to select different types of elements in a spreadsheet, gathering data through an elicitation study in which users demonstrated what they would do to perform different types of selection.

As in typical elicitation studies [191], participants were presented with the effect of an action (*referent*) and asked to demonstrate the interaction steps (*signs*) they would expect to perform to trigger that effect.

3.3.1 Participants

Sixteen volunteers (6 women, 10 men), all right-handed, aged 20 to 46 year-old (average 26.6, median 25), participated in the experiment. All of them reported using a spreadsheet program regularly on the desktop: Microsoft Excel (11), Apple Numbers (2), Google Sheets (5), or Open/Libre Office Calc (5). One of the participants also reported using Apple Numbers on an iPad Pro tablet. Seven participants reported regularly using a digital pen for drawing on a tablet; three for taking notes or annotating documents; one for browsing the Web. Two participants reported infrequent usage of a digital pen for gaming on a Nintendo Console, and four reported never using one.

3.3.2 Apparatus

The experiment ran on a Windows 10 Pro desktop workstation (Intel Xeon CPU/32GB RAM) connected to a Wacom Cintiq Pro display (24", 3840 x 2160 pixels) equipped with a Wacom Pro Pen 2. The software was implemented as a Web application (Figure 3.5), collecting input events using the W3C Pointer Events API [26]. The application ran in the Chromium (v96) Web browser.

3.3.3 Task and Procedure

To avoid considering selections in an overly idealistic and non-ecological context, our study takes into account not only grid-level and value-level actions from Table 3.1, but also includes several additional actions. These actions involve manipulating elements once they have been selected. The complete set of interactions presented to study participants is detailed in Appendix A.1.1.

Participants entered the room, read and signed a consent form that explained the overall goal of the experiment, then filled out a demographic form. They were explicitly told that there was not one unique answer or universal truth to the different questions they would be asked. We then collected their answers for all 28 referents listed in Appendix A.1.1, as described below.

Participants received the following instructions: *"This tablet is tactile. Imagine that you only have your fingers and the stylus (both tip and eraser) to interact*

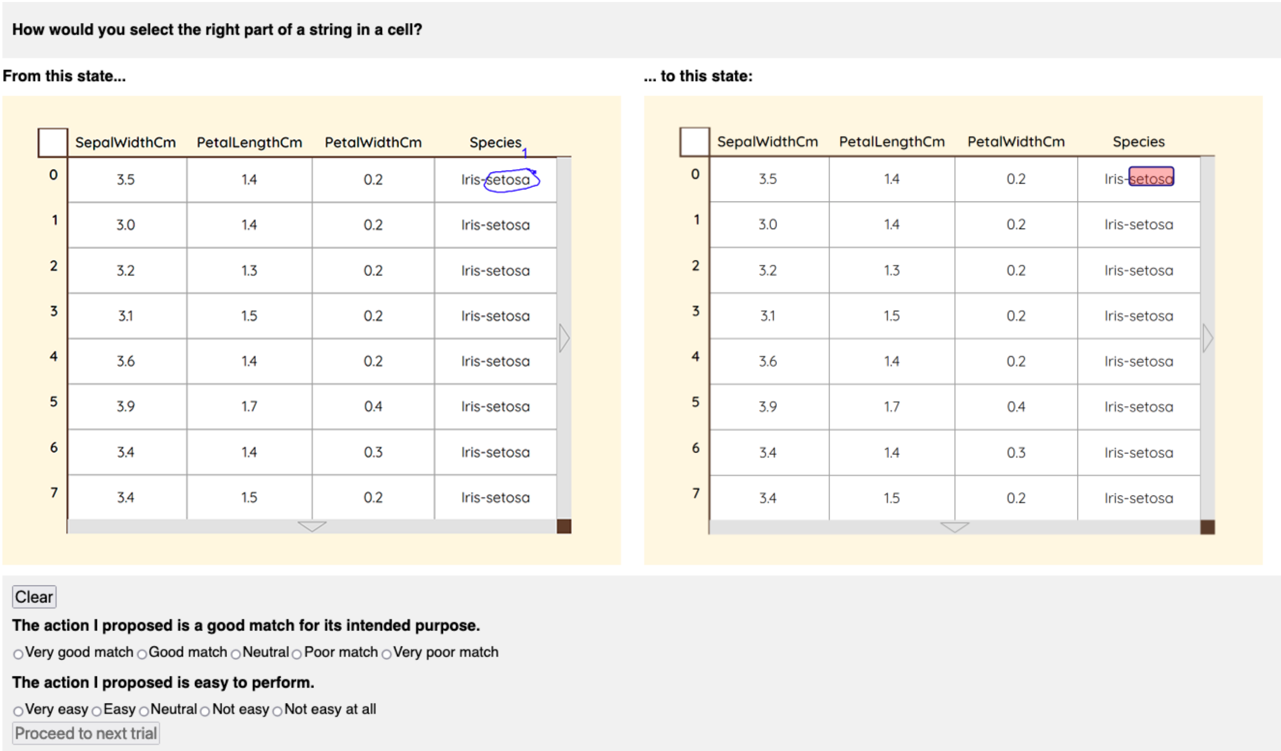


Figure 3.5: A trial in the study. The *referent* is presented as a question in the top panel. The left and right spreadsheets respectively show the state *before* and *after* the referent's effect is applied. Participants use their fingers and pen directly on the left panel to make their *sign* proposal. They then evaluate their proposal with a *Match* and *Easy* score using the 5-pt Likert scales displayed below the spreadsheets.

with it. Use your fingers and/or pen directly on the left picture to show us the action(s) that you would do to trigger that effect. It is up to you whether you perform the gesture using one hand, two hands, the pen only, etc."

In line with studies such as, *e.g.*, [154, 133], that give participants an opportunity to revise some of their signs after having seen the whole set of referents, we divided the experiment into two phases: *Initial* and *Revision*. In each phase, participants completed 28 trials during which they performed actions to invoke one of the referents. Before starting with the *Initial* phase, participants were told that if they ran into conflicts – *i.e.*, if they wanted to perform the same actions (signs) they had performed for an earlier referent – they would have the opportunity to address those conflicts later on if they wanted to and should not worry about this. Once they had become familiar with the complete set of referents in the *Initial* phase, they were given the opportunity to revise any of the signs in the *Revision* phase.

Figure 3.5 illustrates a typical trial in our study. The referent is expressed as a question displayed in the top part of the interface and is illustrated by two pictures of a spreadsheet. The picture on the left shows the spreadsheet's current state and the picture on the right shows the spreadsheet's state after the referent has been invoked. Participants performed the sequence of actions directly on the left picture. As in any elicitation study, the interface was passive and did not implement any spreadsheet-related functionality. It only left digital ink showing when and where input was performed, what type of pointer was used (pen or touch) and what type of actions was performed (tap, dwell, trace). For instance, in Figure 3.5, the participant used the pen to enclose substring `setosa`.

As in Wobbrock *et al.*'s study of user defined gestures for surface computing [191], participants rated how much the signs they proposed were 1) a good match for the referent and 2) easy to perform. They did so thanks to the two 5-point Likert scales at the bottom of the interface (Figure 3.5).

Within each phase (*Initial* and *Revision*), trials were grouped by Scope and Action Type (Appendix A.1.1). The presentation order of groups was counterbalanced across participants using a Latin Square. Within a group, referents were presented in a random order. For each participant, the overall presentation order of the 28 referents was the same across the two phases.

3.3.4 Results

For each trial, the experimental software made a screen capture of the left picture annotated with input traces, and recorded the *Match* and *Easy* 5-pt scores that participants gave to their signs. In case participants iterated over a given trial in the *Revision* phase, the experiment software recorded a revision action and overwrote the data collected in the *Initial* phase with the revised data. In addition, participants were video-recorded and the operator took notes to collect their feedback.

3.3.4.1 Classification into sign and modality categories

Following recommendations by Tsandilas [178], we rely on Fleiss' Kappa coefficient (κ) to assess the level of agreement between participants regarding their proposals for the different referents.

Before computing κ values per referent, we organize participants' proposals into different categories. κ then gives an indication of how much participants agree about the proposed category for a referent. κ's computation takes into account the chance bias to output a normalized value ($\in [-1, 1]$) that can be interpreted as follows: a positive value means agreement beyond chance (+1 meaning perfect agreement), and a negative value means disagreement beyond chance.

We analyze agreement at two different levels: at a fine level by classifying participants' proposals into *sign* categories (*i.e.*, specific series of input ac-

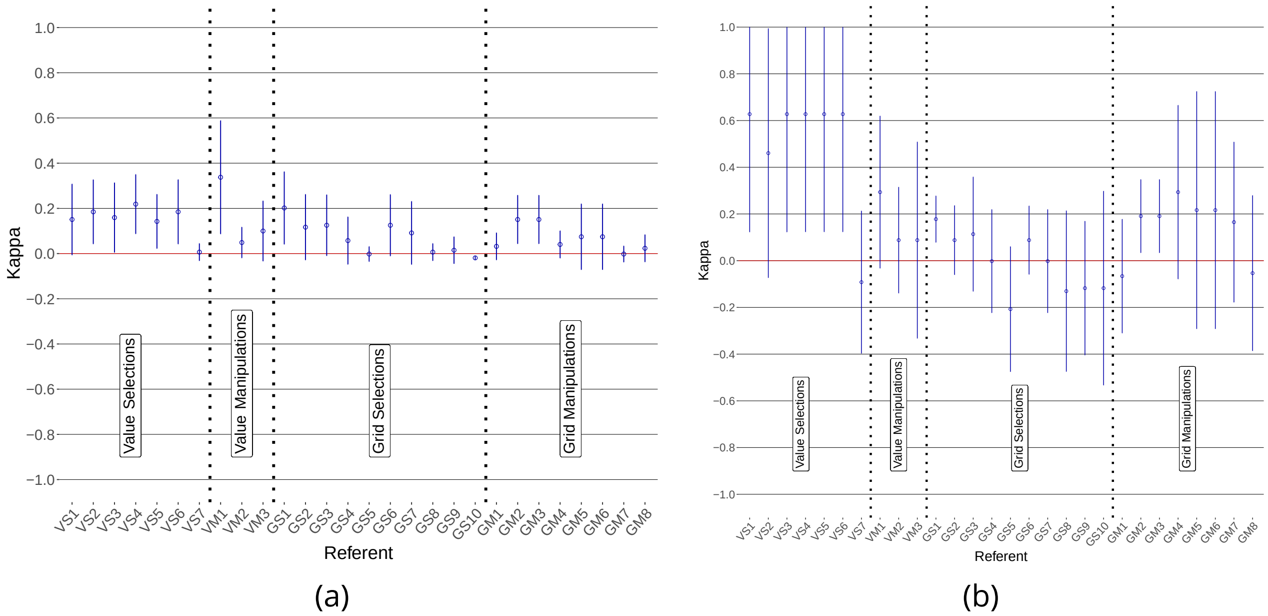


Figure 3.6: (a) Agreement about *sign* proposals per referent. (b) Agreement about *modality* proposals per referent.

tions); and at a coarse level by classifying them into *modality* categories.[3]

3.3.4.2 Level of agreement per referent

The average level of agreement regarding the proposed *sign* per referent (Figure 3.6-a) is low: $\kappa = 0.1 \pm 0.08$ (median=0.1, min=-0.02, max=0.34). The average level of agreement regarding the proposed *modality* (Figure 3.6-b) is only slightly better: $\kappa = 0.18 \pm 0.26$ (median=0.14, min=-0.2, max=0.6). While agreement about modality is low on average, we observe a level of agreement that significantly varies across referents. In particular, a comparison based on confidence intervals regarding the proposed modality shows some difference in κ between groups of referents VS and GS: 0.512 with 95% CI [0.117, 0.906]. Figure 3.6-b illustrates this, showing that κ is much higher for value-level selections (VS) than for grid-level selections (GS).[4] Below we report the general trends observed for value-level and grid-level interaction, by group of referents.

Value-level selections (VS): participants heavily used the pen tip to delimit the scope of a selection within a cell. However, the level of agreement for these referents remains quite low primarily because *i)* some participants (9)

[3]Appendix A.1.2 provides the detailed definition of a *sign* in our study.
[4]Readers interested in the systematic comparison between referents can run analysis scripts provided as supplemental material, which perform such comparisons between pairs of referents and pairs of referent groups.

delimited the scope using a Horizontal Line while some others (6) delimited it by circling it with an Enclose mark; and *ii)* other participants – likely influenced by their experience with current spreadsheet programs – first performed a double tap or dwelled in the cell before inking inside.

Value-level manipulations (VM): moving the selected substring (VM_1) reaches a relatively good level of consensus. All participants but one relied on a Drag-based event. However, they used different modalities: 9 participants used the pen tip, and 6 used their finger. Interestingly, finger drags were usually preceded by a Dwell event. This was likely to make it clear that this drag applied to the value level rather than the grid level. To delete a value selection (VM_2), participants heavily relied on the pen (15) but were split between using the pen tip (7) or the eraser (8). They also used different event types. For example, 2 participants simply dragged the value selection away by dropping it in the background, while others preferred a custom mark to strike through the selection using a Horizontal Line (4) or a ZigZag custom mark (5).

Grid-level selections (GS): the level of agreement is particularly low regarding both the proposed sign and the proposed modality. When selecting a single cell (GS_1), a column (GS_3) or a row (GS_6), the simplest proposals consisted of a Tap event either on the cell, column header or row header. 8 participants consistently made such a proposal but were split regarding the modality: 4 used the pen tip, 3 used their finger while the others used either the pen tip for cell and column selections or their finger for row selections. We observed other proposals such as an Enclose mark around the whole value of a cell to select it (3) or a Flick gesture to select a row or a column. To select a continuous range of cells (GS_2), columns (GS_4) or rows (GS_7), we observed a similar kind of variability but involving a Drag event this time, proposed by 8 participants.

Grid-level manipulations (GM): the level of agreement is low as well. There seems to be stronger agreement regarding the use of the pen modality in comparison with grid-level selections although a statistical comparison is not really conclusive (difference in κ between groups GM and GS regarding modality is -0.155 with 95% CI [-0.405, 0.095]).

3.3.4.3 Match and Easy scores

Match and Easiness scores given by participants to their proposals are reported in Figure 3.7. We use the ARTool package [190, 46] to analyze the effect of *Referent* on Match and Easiness scores. We observe that participants are particularly unhappy with a couple of proposals. VS_7 (*generalizing the selected substring of a cell value to all values in the column*), GS_9 (*select all cells holding a given value in a column*), and GS_{10} (*select all rows holding the same value in a given column*) are given match scores that are overall lower than the other referents. These three referents have match scores that are significantly lower

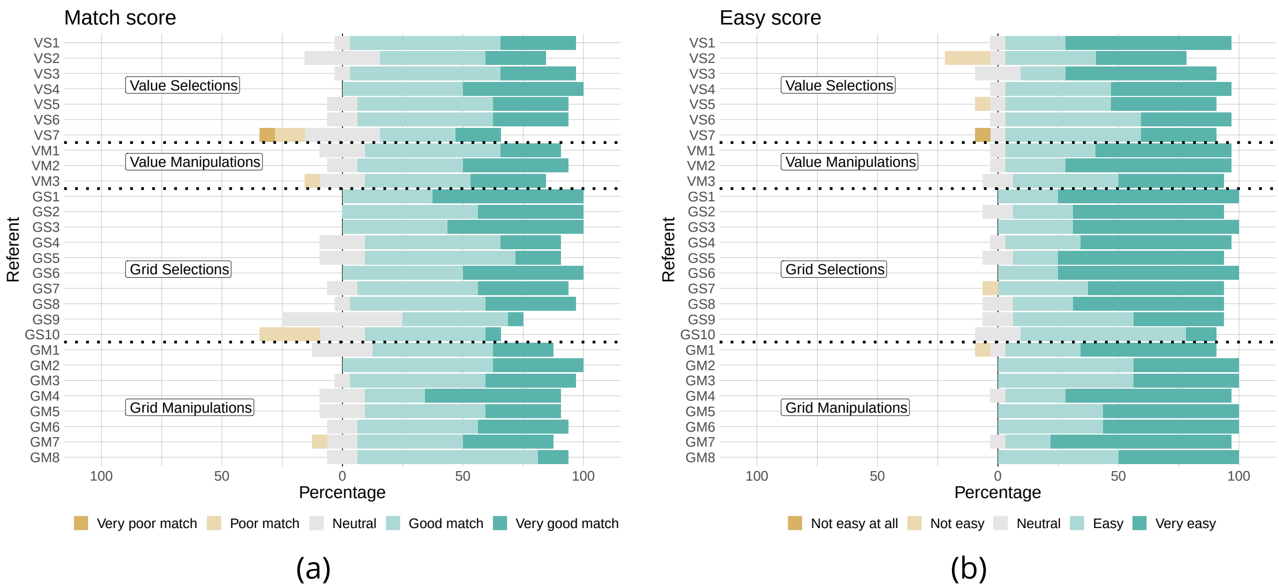


Figure 3.7: (a) Distribution of Match scores per referent. (b) Distribution of Easiness scores per referent.

than that of VS_4, GS_2 and GS_3 ($p < 0.05$). This is not particularly surprising as those three operations are typically not doable with direct manipulations in existing spreadsheet programs. It seems that participants are not able to identify a satisfying way of invoking them even when they are free to choose how to do so. Participants were satisfied with the rest of their proposals overall.

As illustrated in Figure 3.7-b, participants found all their proposals easy to perform. The only exception was GS_{10} (select all rows holding the same value in a given column) whose score is significantly lower than that of many other referents. The difference is significant ($p < 0.05$) when compared to VS_1, VM_2, GS_1, GS_3, GS_5, GS_6, GM_4, and GM_7.

3.3.4.4 Summary of findings

While our participants demonstrated a lot of variability regarding the sign proposals they made for the selections and manipulations considered, our study still yields insights that can inform the design of pen and touch spreadsheet interaction, complementing the first observation drawn from the analysis of commercial spreadsheet programs (**Obs$_1$**, Section 3.2).

- **Obs$_2$**: *The choice of input modality does not disambiguate between grid and value layers.* Participants' proposals do not elicit a clear difference between the pen and touch modalities. The modality alone cannot be used to distinguish between interactions aimed at the grid layer and interactions aimed at the value layer. Seven of our participants actually stated explicitly that they would use one or the other interchangeably for most actions.

- **Obs$_3$**: *The pen is often used for value-level selections and deletion.* Two notable exceptions to the above observation are value-level selections and delete manipulations, which participants systematically performed with the pen. This was likely because of affordances specific to the pen such as its higher precision (useful to point or drag between two characters) and/or the use of pens on paper to circle or underline text. Participants also leveraged their experience with erasers that are often found at the other end of those pens. Most participants turned the pen upside-down and put it on the surface to erase values or grid elements.

- **Obs$_4$**: *Only a few, simple custom marks are used.* A few participants proposed to use a custom mark to activate a command. However, no two participants ever proposed the same mark for a given referent (except the sorting manipulation GS_7 for which two participants used an arrow mark). Furthermore, the set of marks that participants proposed was limited to a small set of simple marks. While such command marks can act as efficient shortcuts for expert users [6], participants did not spontaneously propose to use them.

- **Obs$_5$**: *Pen and touch are seldom used together, multi-touch is almost never used.* Interestingly, there were few proposals that involved pen and touch together. The few instances we observed are for grid-level selections that involve cell-value-matching criteria (GS_9 and GS_{10}), whose specification requires two distinct scopes such as, *e.g.*, a cell value selected with the finger and the header of a column with the pen for GS_9. Finally, the number of proposals that involve multi-touch gestures for selections or manipulations is almost null.

3.4 EunomInk: a Spreadsheet Program Prototype

Taken together, our analysis of pen and touch input in commercial spreadsheet programs and the elicitation study results establish one thing: there is little agreement about how to use pen and touch to interact with spreadsheets. This is true among interface designers (Section 3.2), among end-users (Section 3.3), as well as between designers and end-users.

In this section, we describe the set of interactions that we designed to mitigate frictions between layers. We showcase these interactions using Eu-

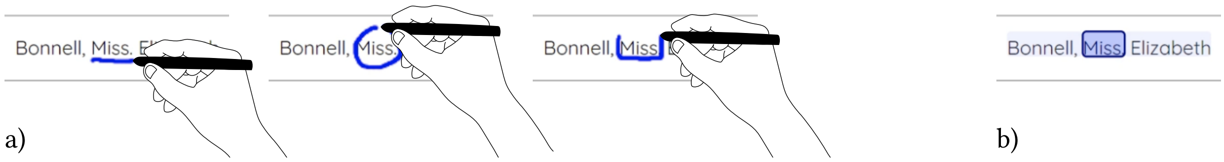


Figure 3.8: Performing a subcell selection by drawing an arbitrary mark: (a) the pen leaves ink, providing an indication of what substring will be selected; (b) once the pen lifted, the ink is replaced by a beautified representation of the selection, consisting of the substring actually delineated (dark blue) and its complements on either or both sides (light blue).

nomInk[5], a prototype Web-based application (whole interface illustrated in Figure 3.15, see Appendix A.2.1 for implementation details). Most figures show data from the Titanic dataset [83]. The companion video also demonstrates most of the interactions described here.

3.4.1 Mitigating Friction Between the Grid and Value Layers

3.4.1.1 Selection

While we did not observe a clear difference between pen and touch for disambiguating between grid-level and value-level selections in the elicitation study ($\mathbf{Obs_2}$), there was a strong tendency to use the pen for value-level selections inside cells ($\mathbf{Obs_3}$). Striving for consistency, we make the choice to have all selections performed with the pen. In order to differentiate a value-level selection from a grid-level selection, we adopt a strategy that takes the graphical context into account [53, 154] to implicitly disambiguate user intent by differentiating input performed *within a cell* from input performed *across cells*, as detailed below:

Grid-level Selection. Basic grid-level selections are performed as follows: a pen tap selects the underlying cell; a pen drag selects the corresponding range of cells; tapping a column or row header selects the corresponding cells. In each case the previous selection is cancelled.

Value-level (Subcell) Selection. In our elicitation study, participants performed different gestures to select only part of a value inside a cell:[6] underline it, circle it, or draw a roughly rectangular shape around it. To support a variety of marks, we adopt a relaxed strategy for substring selection that only involves the inked mark's bounding box: 1) differentiate between

[5] a portmanteau word made of Eunomia, goddess of good order, and Ink
[6] In the remainder of the chapter we refer to this as a *subcell selection*.

value-level and grid-level selection simply based on the position and size of the bounding box (fully inside a cell *vs.* spanning more than one cell); and 2) select the substring of characters that fall within the horizontal range defined by the mark's bounding box. Users do not have to learn any specific mark, as they can use any simple gesture ($\mathbf{Obs_4}$) to delineate the substring. Figure 3.8-a shows three different ways to select a substring. Once the pen lifted from the surface, the system beautifies the selection (Figure 3.8-b): the raw ink is erased and replaced by a set of rectangles that precisely delineate the selection. The most visually-salient rectangle corresponds to the subcell selection. Two low-contrast rectangles, one on each side, correspond to the complements to the main selection and afford some manipulations. Tapping any of these complement rectangles with the pen makes it the main subcell selection, the original one becoming part of the complement. Usage examples of complements are given in Section 3.4.1.2.

The only mark that is confined within a cell and that does not select a substring is a short vertical drag. This specific mark, which has no horizontal range, positions the text caret in-between the two closest characters and enters value-level editing mode.

Once a selection has been made, users are able to extend it in two ways: multiple selection and semantic selection. The scope of these extensions differs depending on whether they are performed at grid level or value level, as detailed next.

Grid-level Multiple Selection. To avoid cancelling the previous selection but rather modify it, users can touch anywhere in the current selection with the non-preferred hand and perform pen taps or drags with the preferred hand. This will add or remove cells to the existing selection rather than reset it. Such incremental selections are typically useful when performing table reshaping operations [71], for instance to extract only a few columns from tables featuring many dimensions. They are available in commercial spreadsheet programs on the desktop but not on interactive surfaces [126].

Value-level Multiple Selection. On the one hand, a *grid-level* selection lets users select multiple cells, but the entire string value gets implicitly selected in each of the cells. On the other hand, a *value-level* (subcell) selection lets users select only a subset of the characters representing a cell's value, but is confined to that specific cell. Our selection model includes a generalization mechanism that enables users to perform a subcell selection in a cell and then apply it across a range of cells. This is particularly useful to edit multiple cells at once, a manipulation that is typically impossible without resorting to

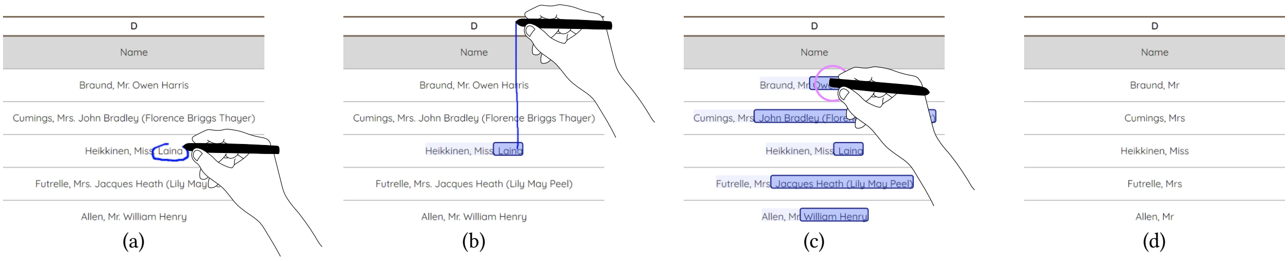
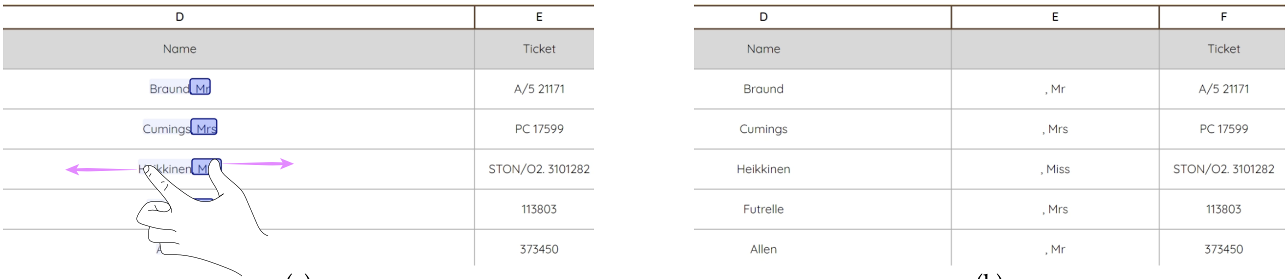


Figure 3.9: Removing a substring across all cells of a column that contains the surname, title and first name of people embarked on the Titanic: (a) circling the end of one cell value using the pen, from the comma separating the title from the first name, to the end of the string; (b) dragging with the pen from the resulting subcell selection to the column header to generalize it to all values in the column; (c) tapping with the pen eraser on any of those subcell selections deletes them all, leaving only the surname and title (d).

scripting tools in any existing spreadsheet program. For example, value-level multiple selection can be used to remove the state code from a list of US cities at once. Users simply have to make a subcell selection in one of the cells, and then drag it over the target range of cells, or all the way to the column's header to apply it to all of its cells – the header representing an abstraction over cells holding data in the column [33]. All matching substrings in range get selected. The value of such a feature would be very limited if it only applied to cells that hold the exact same value. We thus implement an algorithm that infers a pattern from the initial subcell selection and applies it to the other cells. The corresponding generalization algorithm is described in Appendix A.2.2. In Figure 3.9, first names are removed from all cells in a column that contains full names.

Grid-level Semantic Selection. In some desktop word processors, a single-click positions the text caret between the two closest characters; a double-click selects the entire word; a triple-click selects the entire paragraph. Similarly, in our selection model, multiple taps with the pen in the same place trigger semantic selections. A single-tap selects the cell; a double-tap selects all the cells that have the same value in the parent column; a triple-tap selects the lines that those cells belong to.

Value-level Semantic Selection. In the same fashion, a double-tap on a subcell selection selects all matching substrings across all cells in the parent column, and a triple-tap selects the rows that hold matching cells. In cases where the substring to select is the exact same across the range, a double-tap on a subcell is functionally equivalent to dragging the subcell selection to the column header in order to generalize it to the entire column.



(a) (b)

Figure 3.10: Splitting title and surname in two separate columns. After generalization of the person's title selection (from comma to end of string), (a) performing an outward pinch gesture with one finger on the main subcell selection and another finger on the complement splits the two in separate columns (b).

3.4.1.2 Manipulation

Spreadsheets are used for a variety of purposes, and the proportion of *direct data* – as opposed to *derived data* computed using formulas – is significant [45, 10]. The ability to modify these direct data (insert and remove text, rearrange substrings) is a key interaction with spreadsheets, which is paradoxically poorly-supported on interactive surfaces. In this section, we describe the direct manipulations users can perform on selections.

Moving Selections. As all selections are performed with the pen, touch can be used for manipulations. Users can simply drag a selection with their finger to move it, removing the need for a dwell time. At the grid level, touch essentially performs layout transformations: dragging a grid-level selection with a single finger moves the corresponding cells to a new location on the grid. Entire rows or columns can be moved by dragging their header. Copy & paste is supported as well: a single-finger double-tap on a selection copies it to the clipboard, and a dwell pastes it. Finally, users can also sort rows by performing a vertical flick gesture in the column they wish to sort by, as in TableLens [138].

Transforming Selections. The above manipulations involve only one finger. But other manipulations are best expressed using two fingers. An outward pinch gesture performed with one finger on a subcell selection and the other on its complement splits the two parts of the string in separate columns. Again, if the gesture is performed on a generalized subcell selection, all cells get split at once, as illustrated in Figure 3.10. This is useful for instance to split a formatted date (`dd/mm/yyyy`) in three columns [69], or make a landline prefix (`tel:` *vs.* `fax:`) its own dimension [71]. The converse layout transformation – merging two columns – is achieved using an inward pinch gesture.

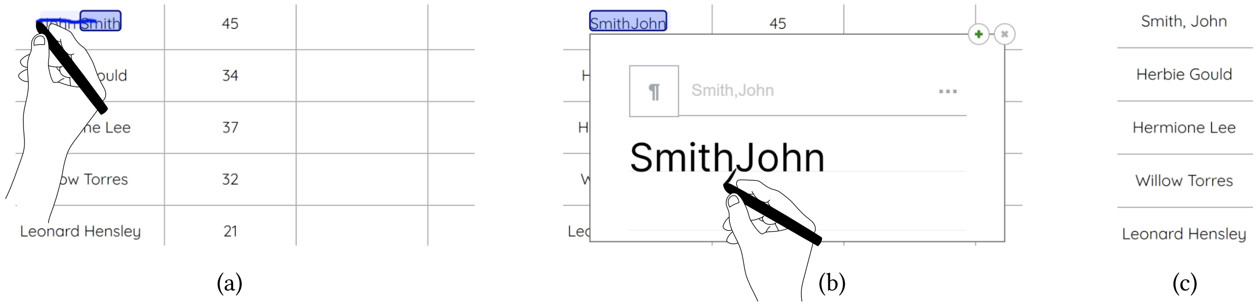
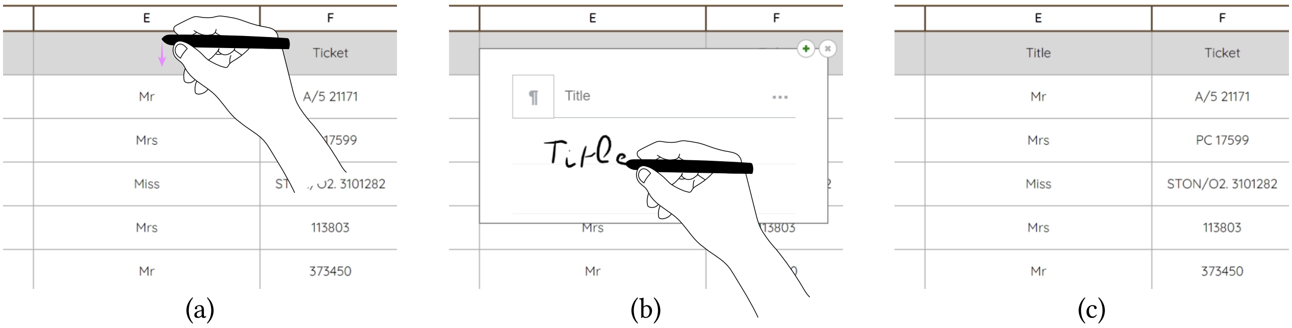


Figure 3.11: Inverting first name and surname (`John Smith` → `Smith, John`) by direct value-level manipulation in one cell: (a) dragging the surname subcell selection to the left; (b-c) inserting a comma followed by a white space between surname and first name.

The touch modality may not always be precise enough when performing value-level manipulations, however. We thus make the choice to enable moving subcell selections with pen drags. The disambiguation between a selection and a manipulation is simply based on the start location of the pen drag: if the movement starts on a selection, it moves that selection and relocates it at the index where it is dropped; otherwise, it initiates a novel selection. As illustrated in Figure 3.11, this makes it easy for instance to invert first name and last name, possibly adding syntactic elements in between, as discussed in Rigel [35].

Deleting Selections. We assign the delete manipulation to the other end of the pen throughout the interface, as we observed many elicitation-study participants turning the pen upside down to erase values (**Obs3**). A tap with the eraser deletes the current selection. At the grid level, this will erase the cells' contents, leaving the grid untouched, unless the eraser taps a header, in which case the corresponding row or column gets deleted. At the value level, the eraser only deletes the selected substring, but possibly does this across rows if the subcell selection has been generalized to multiple cells (Figure 3.9-c&d). The eraser can also remove foreign objects such as annotations (freeform ink and post-it notes, illustrated in Figure 3.15) when they are supported [130].

Editing Selections. Another important value-level manipulation consists of inserting new text or replacing existing text, for instance to fix wrong values in a table as illustrated in ActiveInk [148]. Making a short downward vertical mark in a cell or dwelling on a selection pops up a widget for handwritten text entry (Figure 3.12). Consistent with other manipulations, it is possible to edit multiple cells simultaneously by invoking the widget on a multiple selec-



(a) (b) (c)

Figure 3.12: Value-level editing. (a) Entering value-edit mode by performing a vertical pen drag gesture within the cell's boundaries; (b) handwriting the new cell value which gets recognized and (c) inserted in the cell.

tion. Our prototype uses an external library[7] to parse the ink input by users. It enables the input of new values and the editing of existing cell values as well. It also includes predefined gestures to, *e.g.*, remove a word by scratching it. Handwriting recognition libraries of this type have actually become good enough that they can parse mathematical expressions, opening the door to pen and touch spreadsheet formula authoring, which would be worth exploring as future work.

A few simple syntactic transformations [69] could also be performed directly with short vertical upward pen movements on a selection. For instance, in our current prototype, we map this event to letter case toggling, as this operation is often complementary to the value-level substring transformations introduced earlier.

Most value-level selection and manipulation actions presented in this section are not possible in current commercial spreadsheet programs, even on the desktop. The selection model and manipulation techniques presented here make all of this possible by direct manipulation, relying on simple input actions only. The companion video demonstrates how such capabilities enable advanced editing operations, which involve syntactic, semantic and layout transformations [70] with quick and simple interactions.

3.4.2 Mitigating Friction Between the Grid and Navigation Layers

Our analysis of commercial spreadsheet programs identified some friction not only between the grid and value layers, but between the grid and navigation layers as well (Section 3.2). At the same time, we observed that multitouch

[7] MyScript iinkjs [117] submits the input ink to a Web service and gets the recognized string as a result, with very little latency, providing incremental feedback after every input stroke.

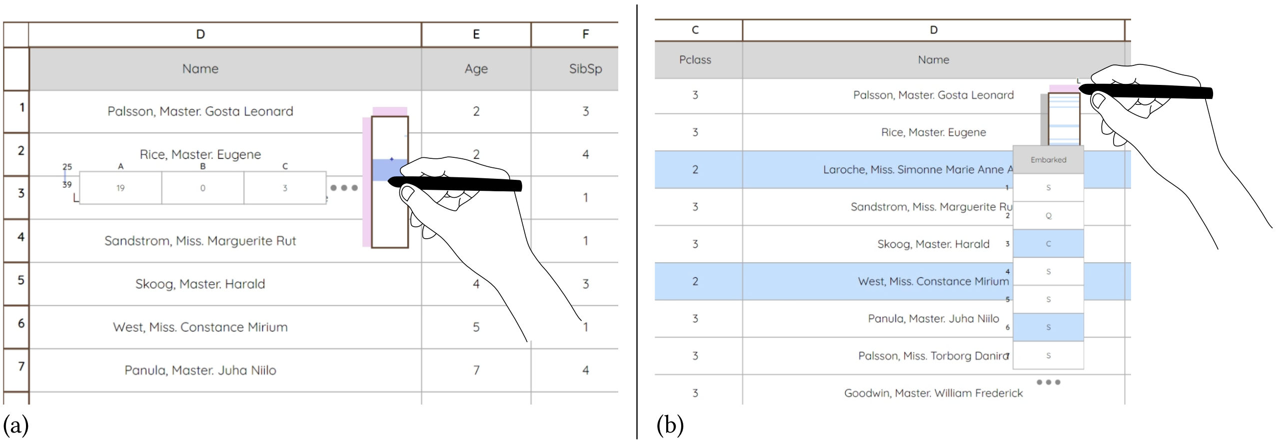


Figure 3.13: Minitable widget for pen selections in large tables. a) Selecting from rows 25 to 39 using the minitable. A preview of the row that falls below the cursor (row 39 here) is visible left of the pen tip. Row selection start and end indices are also displayed. b) Brushing through columns in the gutter above the table shows a preview of the columns (column L here, as indicated next to the pen tip, showing where passengers embarked). Light blue lines in the minitable give an overview of which rows are currently selected (in this example, all rows with Pclass=2).

input was dedicated to navigation in all spreadsheet programs we examined ($\mathbf{Obs_1}$).

In the selection model and manipulation techniques presented in the previous section, we were careful not to introduce any action that would have conflicted with the typical two-finger slide and pinch gestures used for navigation. The few two-finger manipulation gestures mapped to merging and splitting selections are easily disambiguated based on the specific elements they are performed upon. By doing so, we can leverage the only interaction coherently implemented across spreadsheet programs, and minimize friction between the grid and navigation layers.

We aim to further reduce this friction by designing widgets that streamline grid-level selection and navigation. These are relevant primarily to cases where the spreadsheet holds large tabular datasets – as opposed to smaller, more informal spreadsheets with a looser structure. Observing that people often find it easier to utilize their fine motor skills with a pen rather than touch [144], we designed two types of pen-operated widgets, namely the Minitable and the Minivis, that enable advanced grid-level selections and make navigation in the spreadsheet faster.

Minitable. Mapping tables to a small overview, the *Minitable* (Figure 3.13) is a minimap that lets users brush through its rows and columns and select them without having to pan the main viewport. Figure 3.13-a illustrates the selection of a range of rows from the minitable. When dragging vertically

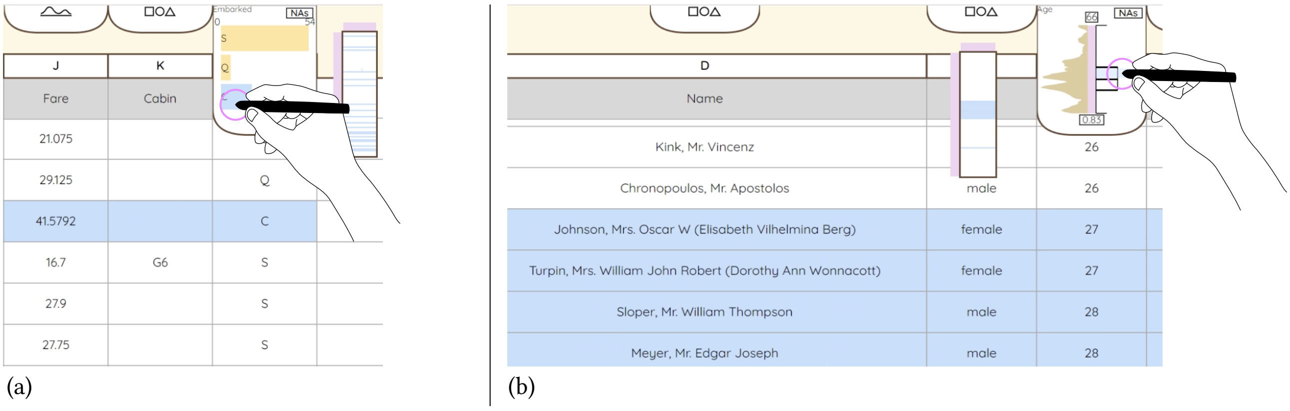


Figure 3.14: Minivis plots. (a) A bar chart shows the distribution of values for categorical variables – here one of three ports where passengers could embark on the Titanic. The blue bar indicates that the user has selected rows featuring C (for *Cherbourg*) as the port of embarkation. (b) A density plot and a box plot are juxtaposed to visualize the distribution for quantitative variables – here the age of passengers. The blue area in the box plot indicates that the user has selected rows featuring an Age value that falls in the third quartile.

with the pen inside the minitable, the system displays a preview of the row that falls below the pen tip. The preview is placed next to the minitable to avoid problems of divided attention. Users can also get a preview of a row (resp. column)'s cells by dragging the pen in the gutter (painted pink) on the side (resp. top) of the minitable, without changing the current selection. Figure 3.13-b illustrates this on columns. Similar to the minimap found in recent code editors, the minitable can be used to quickly scroll to a distant location in the worksheet: releasing the pen while inside the gutter (drag or tap) will automatically pan the worksheet to the corresponding location. Beyond navigation and selection, the minitable enables some manipulations as well. The simultaneous use of pen and touch enables users to move columns or rows to distant locations by direct manipulation: holding a column with the non-preferred hand, users select a distant location with the pen on the minitable and release the column once there. The combined use of pen and touch not only reduces friction between the grid and navigation layers, but actually facilitates a manipulation that was either tedious (on the desktop) or impossible (on interactive surfaces).

Minivis. The minitable provides means to interact with large tables, but essentially allows selections of contiguous blocks of rows or columns, regardless of their contents. It enables *grid-based* selections. Minivis plots (Figure 3.14) are complementary. They enable *value-based* selections. A minivis plot shows the distribution of values in a column depending on its type: a bar chart for categorical columns (Figure 3.14-a); a chart composed of a density plot on the left and a box plot on the right (Figure 3.14-b) for quantitative columns. Minivis plots help characterize the distribution of values [138] and identify rows of particular interest (modes, clusters, outliers [163]). They are interactive, and here again the pen acts primarily as a precise selection tool. Users can perform a variety of value-based selections such as, *e.g.*, *"all rows of a given category"* by tapping on the corresponding bar (Figure 3.14-a); *"values that fall in a given quartile"* by tapping the corresponding region in the box plot (Figure 3.14-b); or a range of values by dragging vertically on the density plot. Minivis plots are similar in spirit to scented widgets [187], but applied to row selection in tables. Pink gutters similar to those of the minitable let users preview distribution values without actually changing the row selection or navigating the table.

Both the minitable and minivis plots leverage the specific affordances of the pen. The high motor precision and limited visual occlusion of the pen tip (compared to finger touch) make it possible to efficiently perform precise selections in spreadsheets holding large tables while keeping the visual footprint of those widgets reasonably small. They decrease the need for navigation actions, which is beneficial for different reasons depending on the interactive surface's dimensions: on a handheld tablet because few rows can be shown at once; on a larger digital drawing board because navigation often requires performing large movements. To further optimize screen real-estate use, minivis plots are only shown on-demand by pulling them from above the corresponding column header.

3.5 Qualitative Study

In the previous section we have seen how we use the expressive power of pen and touch to design interaction techniques that reduce friction between the grid, value and navigation layers. We now report on a semi-structured qualitative study [19] aimed at gathering evidence about users' ability to appropriate those techniques. We developed a prototype (Figure 3.15) that implements all the techniques described in the previous section and supports handwritten annotations as found in [130]. Two configurations among those we examined are now offering this feature on interactive surfaces (dashed cells in Table 3.1). Those annotations typically reside on yet-another layer, and we wanted to see if this might cause unanticipated friction. Our prototype thus lets users create annotations, both freeform ones using a side palette (Figure 3.15-d & e) and post-it notes (Figure 3.15-f & g).

There is much diversity in the expertise of people who use spreadsheets [33], for what purpose they use them, and how they use them [10], ranging from lay users who see them as a way to give some structure to their data to experts who master complex formulas and macros. But we

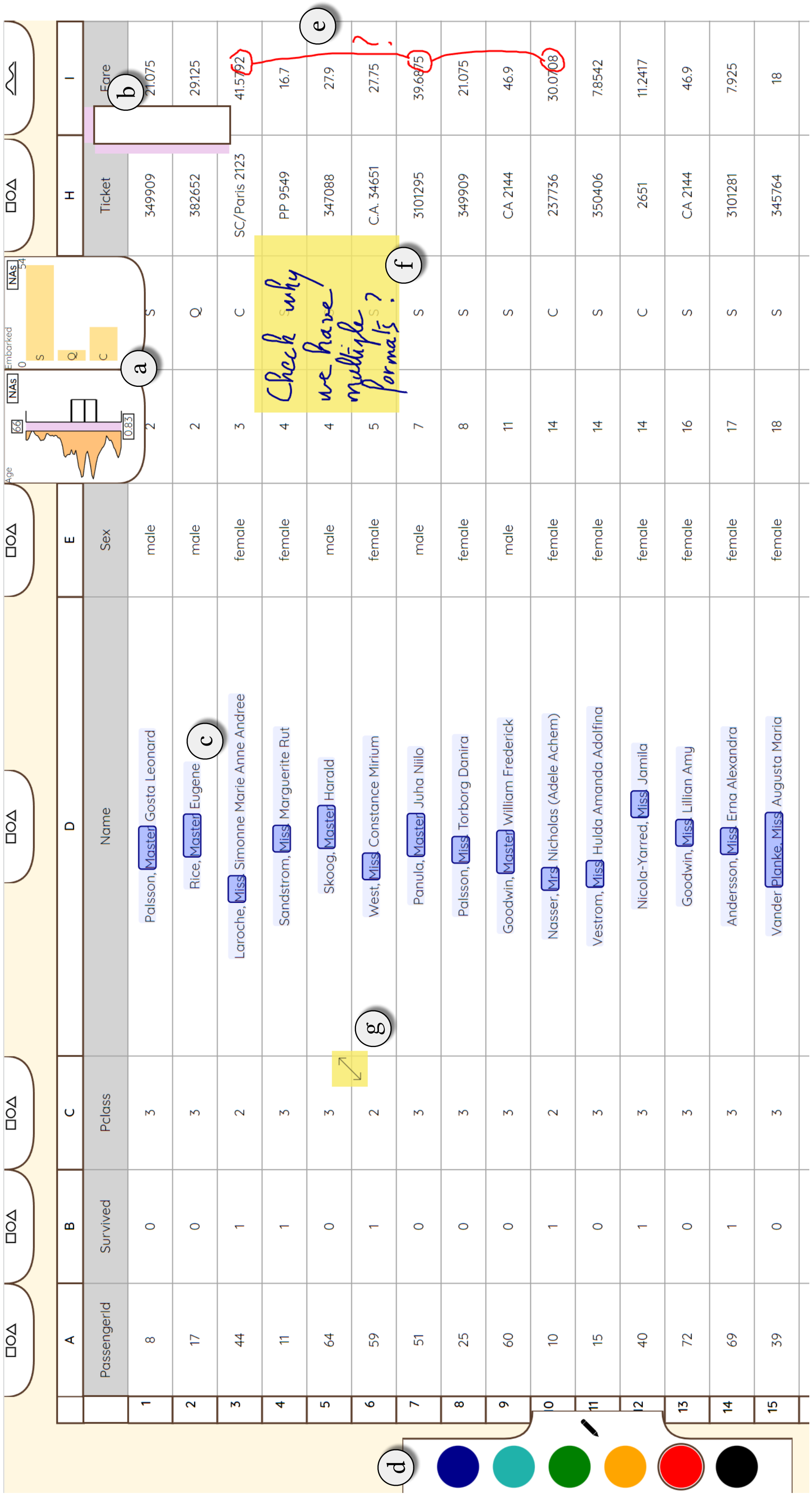


Figure 3.15: EunomInk, the prototype Web-based application used in the qualitative study. Several features are illustrated in this screen capture: (a) minivis plots for row selection; (b) minitable for grid selection and worksheet navigation; (c) substring selection across cells; (d) annotation palette; (e) freeform ink annotation; (f) post-it annotation; (g) post-it annotation minimized.

were primarily interested in evaluating how users appropriate the new direct manipulations and how effective they are at reducing friction. We thus purposefully recruited participants who use spreadsheets to interactively explore and wrangle [86] their data. While such fairly advanced information workers are not the only ones who would benefit from the new techniques, we hypothesized that they were most likely to try them and give meaningful feedback in the relatively short time span (approximately 1 hour) of our semi-structured qualitative study.

3.5.1 Participants and Apparatus

Six volunteers (2 women, 4 men), all right-handed, aged 23 to 34 year-old (average 27.7, median 26.5), participate in the experiment. P1 is a musician; P2-P6 are computer scientists with a specialization in data visualization or HCI. All of them frequently manipulate tabular data but use different tools for this purpose. P1 prefers spreadsheet programs; P6 primarily uses scripting languages; P2-P5 use both spreadsheet programs (Apple Numbers, Microsoft Excel) and scripting languages (R, Python). None of them has used pen and touch to interact with spreadsheets before. P4 and P5 have accessed Google Sheets on a smartphone on a few occasions.

The apparatus used for the study is the same as that of the elicitation study (see Section 3.3.2), but with a display scale set to 200% to compensate for the very high pixel density (3840 x 2160 pixels on a 24" screen). The software is implemented as a Web application (Figure 3.15, see Appendix A.2.1 for details about the technical implementation).

3.5.2 Task and Procedure

Participants entered the room, read and signed a consent form that explained the overall goal and procedure of the experiment. They had been asked to provide a dataset of their own choosing – that they were working on or had been working on recently – several days before the scheduled session in order to check that the dataset worked properly with our prototype. The study then consisted of the following steps.

- A short structured interview ($\approx$ 10 minutes) during which participants were asked about their prior experience with pen and touch devices; experience with data manipulation and use of tabular data in general; the tools they were using and the challenges they were facing when working with data.

- Then the facilitator introduced the system ($\approx$ 25 minutes), demonstrating[8] the possible actions to participants, who reproduced them right after the demonstration, one at a time. The introduction was carried out using the Titanic dataset [83] used in many figures in this chapter. Similar selections and manipulations were grouped together, resulting in 30 short sequences. Throughout this hands-on introduction, the facilitator ensured that the participant had a good understanding of the possible manipulations so far and answered any question they may have had.

- Participants were then presented with their own data and could manipulate them freely. This second phase (34-to-55 minutes depending on the participant) was open-ended. There was no predefined task. Participants explored and manipulated their data at will following a think-aloud protocol. If a participant ran out of ideas too quickly, the facilitator asked questions about their dataset to foster new ideas and manipulations.

- Finally, participants filled out a questionnaire ($\approx$ 10 minutes) to evaluate how useful and easy-to-use different features were. Data were collected using 5-point Likert scales, ranging from *"1. not useful at all"* to *"5. very useful"*; and from *"1. not easy to use at all"* to *"5. very easy to use"*.

3.5.3 Results

While some participants used the system without a clearly-defined goal, others managed to achieve several of their goals, and even made unexpected discoveries about their data, as discussed next.

Figure 3.16-a gives an overview of the think-aloud phase. Color hue encodes the different types of interactions. The rectangle's position, above or below the participant's timeline, indicates that the interaction sequence is positive or negative overall. Sequences were coded as follows. An interaction sequence was considered positive if it led to the intended effect, or if it led the participant to orally formulate a hypothesis, a conclusion, or a positive comment related to it. An interaction was considered negative if it did not lead to the intended effect, or led the participant to orally formulate a negative comment about it. Positive and negative comments about the system as a whole, that cannot be linked to a specific interaction sequence, were coded as general comments.

Figure 3.16-b visualizes the relative time spent performing manipulations of different types: grid-level basic operations; value-level direct manipulations; minitable and minivis selections; annotations. The legend also indicates the ratio of positive *vs.* negative sequences for each type.

[8]The script used for this introduction, as well as data analysis scripts and post-hoc questionnaires, are available as supplemental material.

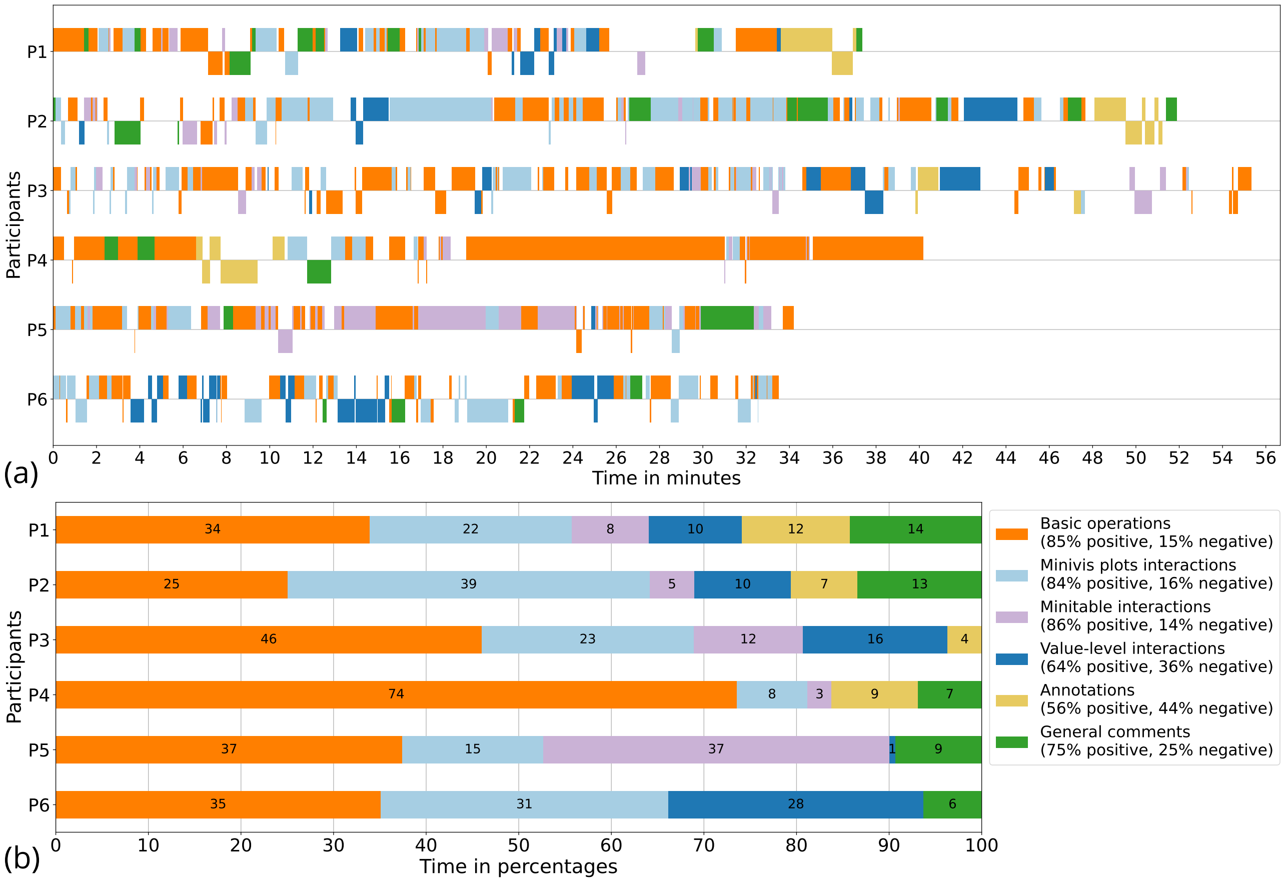


Figure 3.16: (a) Interaction sequences of the think-aloud phase, with one timeline track per participant. (b) Relative time spent performing interactions of different types, per participant.

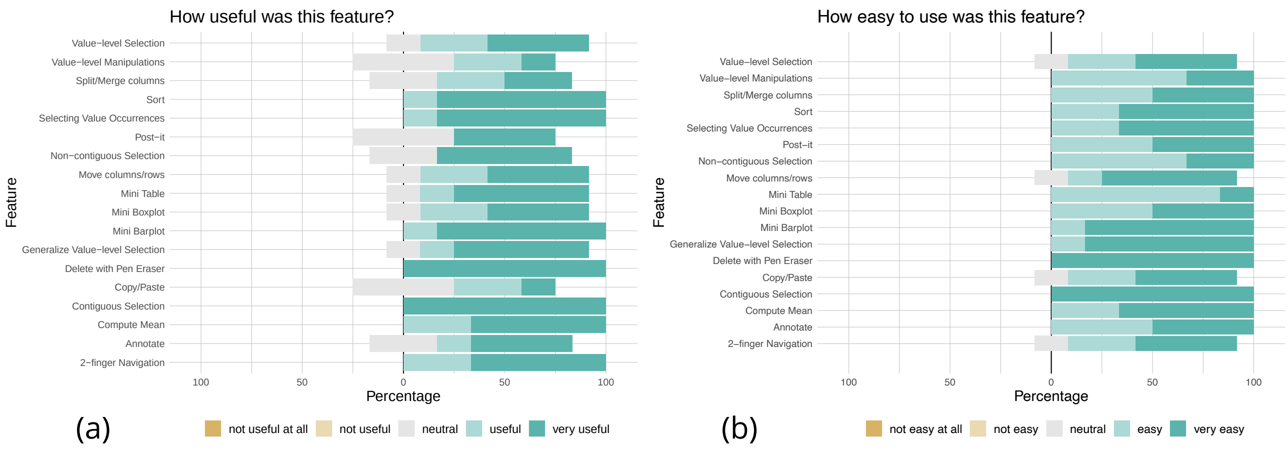


Figure 3.17: Participants' evaluation of (a) usefulness and (b) ease-of-use for 18 interactive features using 5-point Likert scales.

3.5.3.1 Basic Operations

Unsurprisingly, basic operations (navigation, simple grid-level selection) amount to a significant proportion of all manipulations and are intermingled with other types of interactions. Participants were generally satisfied with them (85% positive).

All participants used two-finger gestures to navigate the worksheet, in line with $\mathbf{Obs_1}$. Combined with selections to highlight particular values, panning the worksheet helped relate data points that were too distant to be shown simultaneously on screen. For instance, P5 was able to make sense of some data points that they had initially considered as outliers. They also identified patterns within selections to formulate hypotheses on their data. When navigation involved too much back and forth, some participants (P1, P2, P4 and P6) rather chose to perform a layout transformation, regrouping the columns of interest with basic single-finger drags. Bringing those columns together also enabled P2 and P6 to compare their distributions side-by-side thanks to the minivis plots.

P1 and P6 made use of two-finger pinch gestures to quickly merge or split columns. For instance, P1 fixed values that were wrongfully encoding different categories. P1 used column merge to create a new categorical column from two others, commenting that *"it enabled me to create new correlation data that I could then plot and compare."* The ability to sort values quickly with a single-finger flick on a column was also well-received by P3, P4 and P5 who used it many times to successively sort columns either for exploratory purposes (P3, P4) or for arranging their table according to a specific sorting they had in mind.

In line with $\mathbf{Obs_3}$, participants also appropriated the pen eraser quickly, rating it 5 on the usefulness scale (Figure 3.17-a). Using the pen for selection and then the eraser for deletion, P2 removed all rows but the ones with a specific value in one column, before deleting that column.

Semantic selections, which are difficult to achieve with regular spreadsheet programs, were found useful on multiple occasions. For instance, P3 wanted to know if a particular value they had found was unique, and performed a pen double-tap on it to look for other instances. Conversely, P1 used the same interaction to check that multiple values they had edited were indeed still equal. They also used it to replace missing values by zeros in an effective way: first, input a '`0`' with the pen in an empty cell and copy it with a finger double-tap, and second, perform a semantic selection with a pen triple tap on one of the remaining empty cells to select all of them, and then paste. P2 and P5 used the pen triple tap to select all rows with a matching value in a categorical column of interest. P5 observed that *"The triple tap is great, it allows to do things quickly."* Overall, participants rated these interactions very positively (Figure 3.17): 4.67 (easiness) and 4.83 (usefulness).

3.5.3.2 Value-level Manipulations

Subcell selections and manipulations do not exist in commercial spreadsheet programs. Yet participants appropriated them quickly to perform elaborate transformations by direct manipulation. We first report on how they approached the selection step, and then detail the purpose of their value-level manipulations.

Subcell Selection. Participants delineated selections in different manners, confirming the need to support arbitrary marks for this purpose (Figure 3.8): P1, P2 and P5 systematically circled substrings, whereas P3 and P6 systematically underlined them. We did not observe any confusion between grid-level and value-level selection – both performed with the pen – again in line with $\mathbf{Obs_3}$ and suggesting that our disambiguation strategy (Section 3.4.1.1), while very simple, is effective. Participants always applied subcell selections to a group of cells, never to a single cell only. *Multiple subcell selections* and *semantic subcell selections* thus played a central role in value-level manipulations. P5 performed semantic subcell selections, using a triple tap in order to select entire rows with a cell matching the selected substring.

Purpose: Table Reshaping. Some subcell selections were used to split columns. For instance, P6 wanted to split a column holding `"name, date"` values in two columns. They selected the date substring in a cell and generalized that selection to the entire column, obtaining two selection groups: the dates and their complements. They then only had to pinch outward, one finger on each group, as illustrated in Figure 3.10. Realizing that there were trailing commas after the names, they selected one, generalized it, and used the pen eraser to delete them all. P1 also split a column, which was holding `"foo - bar"` values. But they removed the dashes thanks to a semantic subcell selection double-tap as they were confident that this was the sole delimiter used in all values.

Purpose: Data formatting. Multiple subcell selections were also used to perform syntactic transforms [69]. P2 and P3 used them to format numbers in quantitative columns. P3 knew that all values had the same number of fractional digits and wanted to keep the first one only. They performed a right-side generalization with an (implicitly-fixed) number of characters and erased it. P2 had a similar intent, but in their case the count of fractional digits varied from cell to cell and they adopted a different strategy. Aiming to only keep the first three digits, they selected them in a cell, generalized that selection, and then inverted it by tapping on the complement, effectively selecting all trailing fractional digits beyond the first three (when any). They actually had to perform this transformation in several columns, and wished it were possible to generalize subcell selections to multiple columns at once, suggesting this might be achieved by dragging the pen across target column headers, explaining that *"It would do the same type of selection on the other columns, because I know that they share the same properties."* P2 also relied on a semantic subcell selection to replace decimal points by commas. Tapping on the selection of all points with the pen, they opened the ink text editor, scratched the point and wrote a comma instead, effectively replacing them all in a matter of seconds.

P1 and P6 combined subcell selections with simple gestures (**Obs4**) to format columns containing string data. The small upward stroke gesture from a subcell to change letter case gave them *"the feeling to uplift the selection."* They found it easier to make this sort of edit than with other spreadsheet programs: *"Once I have formatted a cell with the proper casing, I copy/paste it to all other cells that need to be fixed. It's very tedious."* P6 went further with the manipulation of their selection: they dragged it from right to left, effectively swapping it with the complement. As the letter case was now wrong, they changed the first characters to uppercase, and the middle characters – formerly the first characters – to lowercase, all with a short series of swift pen gestures.

Purpose: Data editing. Finally, we observed multiple occurrences of raw data editing. For instance, P2 – who had inadvertently deleted the content of a cell – invoked the ink text editor with a short vertical drag in that cell and rewrote the string by hand. As mentioned earlier, P1 wanted to replace all

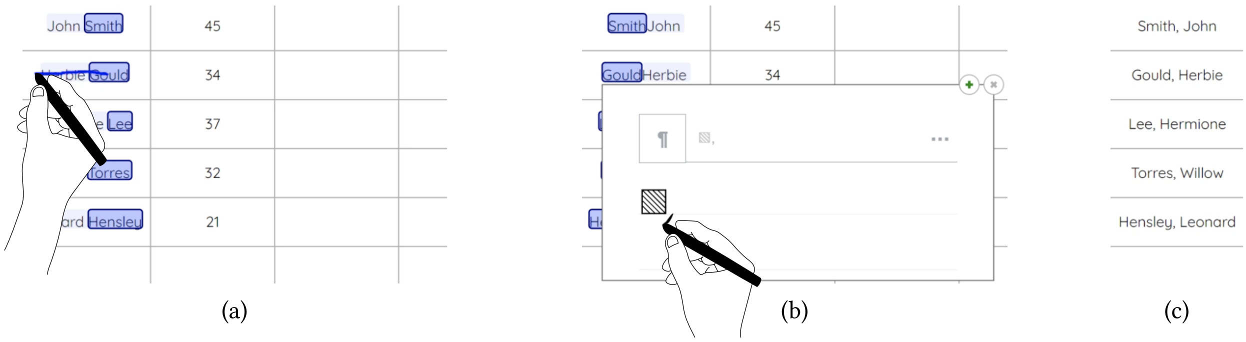


Figure 3.18: Inverting first name and surname (`John Smith` → `Smith, John`) by direct value-level manipulation as in Figure 3.11, but for all cells at once: (a) dragging the generalized surname subcell selection to the left; (b-c) inserting a comma followed by a white space between all surnames and first names.

missing values by zeros and achieved this by writing only one `'0'` and copying it at once in the remaining cells thanks to a semantic selection. P3 performed a similar syntactic transformation, replacing all second words in a range of cells by another word written with the pen.

Some participants tried to achieve more advanced editing on multiple subcell selections and highlighted one missing capability. The current interaction model allows erasing, moving, and replacing substrings. But in the latter case matched substrings can only be replaced by one unique new substring – written with the pen – across the range. For instance, coming back to the example in Figure 3.11, inverting first name and surname for all rows at once can be done with a multiple subcell selection (performed before the action depicted in Figure 3.11-a). But inserting the comma and whitespace between surname and first name is not straightforward: invoking the ink text editor on surnames would then replace the surname in all cells with the same input value, irrespective of the original value. What is missing[9] is a conceptual equivalent to text caret generalization, which is somewhat similar to the multi-line editing cursors featured in modern code editors that enable inserting a string at possibly different positions on multiple lines at once.

We evaluated this issue as critical and iterated on our subcell manipulation techniques to address it. As illustrated in Figure 3.18, we added support for editing multiple *heterogeneous* subcell selections by visually representing subcell selections in the handwriting recognition widget. When invoked on a multi-subcell selection whose individual values differ, the widget symbolizes the block using a square glyph (Figure 3.18-b) which act as placeholders,[10]

[9]The absence of this capability, and the fact that the handwriting recognition algorithm often confused 'o' with 'o', account for many (36%) of the value-level interactions categorized as negative (Figure 3.16).

[10]This glyph would ideally look similar to subcell selection rectangles (blue, rounded

leaving the user free to insert characters before and after it at will. The value manipulation described in Figure 3.11 for a single cell can now be performed on multiple cells at once, even if they hold different values.

3.5.3.3 Minitable and Minivis Plots

As they use spreadsheets primarily for tabular data exploration and wrangling, study participants were naturally interested in the minitable and minivis plots (Figure 3.16-a), except P4 . Some participants used both the minitable and minivis plots (P1,P3), while other participants primarily used the minitable (P5) or the minivis plots (P2,P6). P2 and P6 actually interacted very often with minivis plots, performing as much or even more minivis interactions than basic operations (Figure 3.16-b).

Participants used the minitable to navigate the worksheet, sometimes to reach a particular row or column quickly, but most often to move to a specific area regardless of its contents. The same type of interaction was used to quickly go back to the first row, first column, or both. The minitable was also used for some selections that would otherwise have been tedious because of grid *vs.* navigation friction. For instance, P2 brushed over the pink gutter with the pen to locate a precise row label, and continued dragging the pen inside the minitable's body to select a large number of rows from that point. The spontaneous use of pen + touch together was also observed on a few occasions ($\mathbf{Obs}_5$). P5 used the minitable to drag & drop a row to a distant location in the worksheet by holding it with their non-preferred hand and quickly panning the worksheet with the pen on the minitable.

3.5.3.4 Annotations

Participants could annotate the spreadsheet by activating a dedicated side palette to make freeform annotations on a layer above the grid, or by drawing post-it notes that they could write upon and contract (Figure 3.15). However, participants did not use annotations much, either because it is not a regular part of their workflow, or because it was not particularly relevant in the context of a 30-to-45-minute analysis performed during a study.

Participants still provided feedback about this feature. They considered the freeform ink marks and post-it notes useful (rated 4.17, resp. 4.0) and easy to use (both rated 4.5) – see Figure 3.17. Figure 3.16-a shows a large number of annotation-centric interactions with a negative outcome, however (44%). This seems to be mostly due to the fact that annotations did not fully meet the expectations of participants who actually tried them. While most of these issues do not strongly relate to our interaction techniques, one seems worth

corners) to better convey their role, but the iinkjs widget used for handwriting recognition does not support rendering arbitrary graphics out of the box.

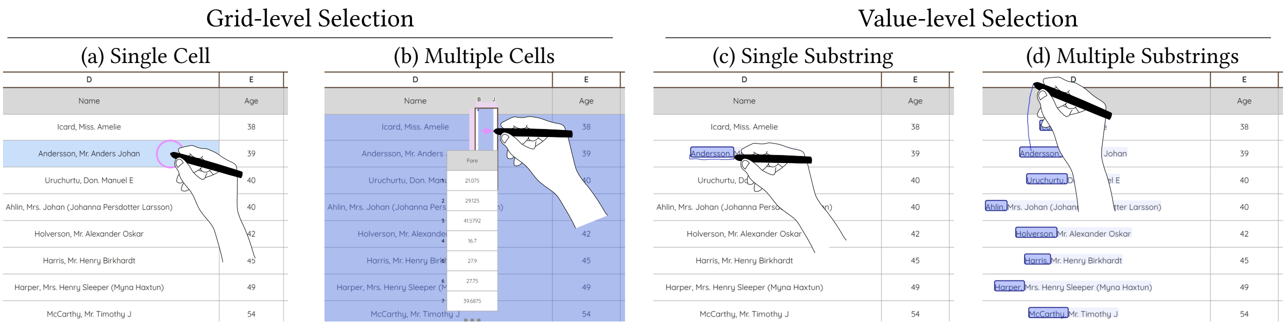


Figure 3.19: Our techniques enable seamless direct manipulations of diverse elements in a spreadsheet primarily by enabling a broader variety of selection actions than existing spreadsheet programs. With only one or two pen marks, users can select different types of scope. Representative selections at the grid and value levels: a) a single cell; b) large sets of cells without navigation – here columns `B-J` extending far beyond the current viewport; c) a substring within a cell; d) multiple substrings matching a pattern across cells in a column.

mentioning. P4 wanted to have a place where annotations would always remain visible regardless of what part of the spreadsheet was visible – *"I want to have them always within reach"* (P4). They were in essence asking for a different annotation layer that would not be tied to the spreadsheet navigation layer. Simply adding another layer would likely cause even more friction, but as suggested by P4 this could be achieved by having a dedicated area in the spreadsheet's periphery that does not interfere with the grid – at the cost of the amount of screen real-estate dedicated to the display of the spreadsheet itself.

3.6 Conclusion

RQ1: How can pen and structure be articulated in order to promote efficiency on interactive surfaces?

The research question driving our investigation was RQ1 applied to the case of spreadsheet programs: how can pen and structure be articulated in order to enable efficiency with spreadsheet programs on interactive surfaces?

Starting from the observation that even simple spreadsheet manipulations can be cumbersome when performed on interactive surfaces, we aimed to analyze the causes of the user experience deterioration and identified a particularly important one: *the grid, while being key to many interactions, acts as a layer that covers the entire workspace and hinders simple manipulations of the cell values that lie below it*. In addition, because interactive surfaces are primarily operated using direct input, spreadsheet programs that run on them actually feature a third layer dedicated to navigation actions that also covers the entire workspace. All these superimposed layers create ambiguity regarding the intent of many input actions, requiring users to perform cumbersome interaction sequences to reach the intended target elements and reduces efficiency. Having identified this key issue, we proposed a set of interaction techniques that help **bypass the structure when needed**: they leverage the precision and expressive power of the pen to disambiguate between layers, letting users *break through the grid* and seamlessly select a variety of elements with simple pen-based actions.

We first examined a representative sample of commercial spreadsheet programs running on a range of operating systems and pen and touch hardware (see Section 3.2). This highlighted where inconsistencies and consensus across hardware and software configurations lay. We then conducted an elicitation study to better understand users' expectations when using pen and touch to interact with spreadsheets (see Section 3.3). Participants achieved little consensus, but the study yielded insights about the role of pen and touch as they perceive them, and the sort of gestures they would like to perform for direct manipulations. We then built upon those insights and our own experience as interaction designers to propose a new selection model (see Section 3.4). Building upon Pfeuffer *et al.*'s division of labor which states that *"the pen selects, touch manipulates"* [130], our model dedicates the pen to selections, but it also analyzes the spatial context of pen marks in order to implicitly disambiguate between the grid and value layers. We showcased how this model supports advanced spreadsheet operations with simple direct manipulation techniques. The efficiency of these techniques was considered at every step of our design process, and we evaluated the usability of the system as a whole by conducting a semi-structured qualitative study involving six information workers manipulating their own data (see Section 3.5). Results from this study suggest that despite the diversity of datasets and manipulations witnessed, people who use spreadsheets for data exploration and wrangling can easily appropriate the new direct manipulation techniques and use them efficiently.

3.7 Future Work

The use of pen and touch for direct manipulation in spreadsheets opens multiple avenues for future work. Some are targeted at all users, others are rather targeted at information workers such as our study participants, who frequently perform advanced data transformations. In that respect, our work can be seen as an early contribution to the emerging topic of post-WIMP interfaces for data manipulation, as discussed by Lee *et al.* [94].

One general avenue for future work is to investigate how to enable additional operations by direct manipulation without compromising usability. Chalhoub & Sarkar emphasize the key role of columns as a *"user-centric structure"* [33] beyond sorting and filtering, observing in their interviews with spreadsheet users that they can be appropriated for other operations including conditional formatting and formula authoring. Some of these *"column-based operations"* could lend themselves to specification by direct manipulation with pen and touch. Advanced transformations, including rearrangement and transposition [73] could be considered as well.

One particularly interesting issue to address is that of formula authoring, which is often tedious on interactive surfaces. As hinted at in Section 3.4.1.2, handwriting recognition libraries such as the one used in our prototype can now parse reasonably-complex mathematical formulas. This means that the pen can be used to manipulate not only plain values but spreadsheet formulas as well. But more interesting opportunities lie in the articulation between selection actions and formula authoring. As both grid-level selections and subcell selections are performed with the pen, the insertion of function calls and selection ranges could both be performed seamlessly with the pen. This would streamline an operation that often requires two devices even on the desktop, as cell selections are typically performed with the mouse while function calls are inserted with the keyboard.

Complementing this, subcell selections can be seen as transpositions to direct manipulation of spreadsheet text extraction functions (such as, *e.g.*, `LEFT()` and `RIGHT()`), combined with basic substring expression synthesis [69]. The visual reification of what essentially comes down to substring selection across cells does not have to be confined to the set of manipulations described in this chapter, however. Subcell selections could be involved in formulas, as discussed above, but they could also be helpful in programmatic approaches to spreadsheet data manipulation [91, 156] where a variety of syntactic, semantic and layout transformations [161] get exposed as short programs.

Finally, mapping more operations to direct manipulation actions with regular pen and touch might prove challenging as few simple gestures remain available. Advanced operations – such as table pivoting, folding and unfolding to name a few – are arguably performed less frequently, mainly by expert users, and could be mapped to input capabilities that have only recently been investigated. These include the combination of pen, gaze and touch [127, 128]; the detection of hand posture [28, 204, 109], of tablet orientation [147]; the use of pen roll events, of passive surfaces to enable interactions outside the worksheet's bounds [100]; and the non-preferred hand's contact shape when working on a large interactive surface [112]. None of those capabilities have been considered in our interaction techniques as we purposefully limited ourselves to capabilities featured in off-the-shelf hardware. As some of them mature and eventually become widespread, it would be interesting to study how to integrate those newer capabilities, comparing them in terms of usability and efficiency to more conventional approaches such as exposing the same functionality *via* contextual menus.

Chapter 4

Breaking the Structure for Creativity

This chapter is based on two full papers written in collaboration with Catherine Letondal, Emmanuel Pietriga and Caroline Appert. The first one, "Challenges of Music Score Writing and the Potentials of Interactive Surfaces" [32] is published in CHI '24: Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems. The second one, "EuterPen: Unleashing Creative Expression in Music Score Writing" [31] is published in CHI '25: Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems. The demonstration video of the prototype is hosted in the ACM Digital Library.

Composers use music notation programs throughout their creative process. Those programs are essentially elaborate structured document editors that enable composers to create high-quality scores by enforcing musical notation rules. They effectively support music engraving, but impede the more creative stages in the composition process because of their lack of flexibility. Composers thus often combine these desktop tools with other mediums such as paper. In this chapter, we first interview nine professional composers to understand their thought process and creative intentions, and rely on the "Cognitive Dimensions of Notations" framework to capture the frictions they experience when materializing those intentions on a score. We then discuss how interactive surfaces could increase flexibility by temporarily breaking the structure when manipulating the notation. Building on this, we introduce EuterPen, a music notation program prototype that selectively relaxes both syntactic and structural constraints while ensuring an efficient workflow. Composers can input and manipulate music symbols with increased flexibility, leveraging the affordances of pen and touch. They can make space on, between and around staves to insert additional content such as digital ink, pictures and audio samples. We describe the iterative design process that led to EuterPen: prototyping phases, a participatory design workshop, and a series of eight interviews. Feedback from the participating professional composers indicates that EuterPen offers a compelling and promising approach to music writing.

4.1 Motivation

Music notations provide composers with a way to represent visually what they have in their mind. The modern staff notation in particular is used in a broad variety of genres. It lets composers share their work with other musicians, and do so with minimal ambiguity. But for many composers the notation also plays a key role throughout the creative process. It is a means for them to capture early ideas, to expand and to iterate on what they have already written. When composing on a computer, music notation software are thus not only music *engraving* tools, but actual creativity support tools that composers use for a variety of activities [151], from the capture of *"germinal ideas"* [14] to the communication and archival of the final piece.

Music notation software can be seen as elaborate structured document editors that share many similarities with visual programming languages [118]: music symbols are essentially graphical primitives structured according to constraints captured by a multi-dimensional grammar [62]. As such, those software enable the creation of *syntactically* well-formed compositions. By enforcing the notation's syntactic rules and automatically laying out symbols to ensure good readability, those programs produce high-quality scores that are ready to be engraved. But at the same time, enforcing those rules impedes the more creative stages of the composition process. These stages rather call for flexibility in the interactive manipulation of symbols and, beyond symbols, of other types of contents that might be relevant to insert in a score during the composition process. These include freeform annotations, pictures, audio samples, foreign score fragments used for reference only.

This lack of flexibility often leads composers to use not only software but paper as well [96], more particularly in the early stages of the creative process [179]. Composers may face *"abrupt shifts"* [57] when switching medium, however. This will sometimes lead them to keep editing on the computer even if paper would be better suited for a particular activity.

The HCI literature suggests that interactive surfaces that support pen and touch input – such as, *e.g.*, the Apple iPad or the Microsoft Surface Pro – could combine the best of both paper and computer mediums to resolve tensions between the need for structure and the need for flexibility (see [30, 150] for recent examples). But while a couple of commercial music notation applications have been designed for interactive surfaces (for instance StaffPad [166] and Symphony Pro [134]), they mostly adhere to the same WIMP-oriented interaction model as desktop applications, treating pen and touch as generic pointing devices such as mouse and trackpad. We believe that this approach is suboptimal, and advocate for a more comprehensive redesign of the whole interaction model to reveal the full potential of interactive surfaces. Such a design process requires close observation of composers' creative processes and an analysis of previous work that has leveraged the distinctive properties inherent to interactive surfaces for other application domains.

This chapter reports on this design process, in which we initially interviewed nine professional music composers following a semi-structured qualitative study protocol [19]. Our goal was to understand their creative process, paying particular attention to how they use music notation software. We wanted to understand what challenges they face, and how they address them. We first report high-level observations to capture their thought process and creative intentions. We then perform a detailed analysis, guided by the "Cognitive Dimensions of Notations" framework adapted to music [118] in order to capture the challenges they face when materializing those intentions. Informed by prior research on interactive surfaces, we discuss guidelines exploring their potential to address the usability challenges previously identified, and what opportunities they provide to better support composers' creative process.

Building on these guidelines, we developed EuterPen[1], a prototype music notation program designed specifically for interactive surfaces. We describe our methodology, which consisted of design and software prototyping phases, a workshop, and interview sessions with eight professional composers. We then introduce EuterPen, organizing the discussion according to how we break away from the rigidity of regular music notation programs, providing composers with advanced editing capabilities based primarily on direct manipulation with a combination of pen and touch input.

4.2 How Composers Think and Work

We conducted 9 interviews with music writers in order to understand their creative process and the challenges they face. Our only selection criterion for interviewees was that they had to write down their music on scores (as opposed to, *e.g.*, using a Digital Audio Workstation). Our interviewees have a variety of profiles. Although all of them have a classical background and received academic training in music theory (8 from the Conservatoire National Supérieur de Musique et de Danse de Paris and 1 from Sorbonne University), Table 4.3 reports on how those different profiles vary in *i)* their music writing activity; *ii)* their musical style; *iii)* their music writing experience; *iv)* the music notation software they use; and *v)* their main writing tool.

The interviews were semi-structured and composed of 6 blocks of questions: *composer profile, writing process, software use, difficulties encountered,*

[1] a portmanteau word made of Euterpe, the mythological greek muse presiding over music, and Pen

Interviewee	Initials	Activities	Main Style	Experience (y)	Current Software	Main Writing Tool(s)
Lucius Arkmann	LA	C*, A*, E*	Modern classical	> 15	Musescore	Laptop or paper
Alexandre Olech	AO	C*, A*	Modern classical	> 15	Sibelius	Desktop
Gabriel Feret	GF	C*, A*	Modern classical	$\simeq 18$	Sibelius	Laptop and paper
Maxime Senizergues	MS	C*	Progressive rock	10	Musescore	Laptop
Philippe Gantchoula	PG	C*	Modern classical	30	Finale	Desktop
Denis Ramos	DR	C*, A*	Modern classical	$\simeq 10$	Sibelius	Laptop and paper
Jérémy Peret	JP	A*	Modern classical	> 20	Finale	Laptop
Coralie Fayolle	CF	C*	Choral and orchestral	> 35	Finale	Laptop and paper
Gustave Carpène	GC	C*	Modern classical	12	Sibelius	Laptop and desktop

Table 4.1: Profile of the nine composers interviewed. *Activities: C=Composition; A=Arrangement; E=Engraving for another composer.

mental model and *handling of musical motifs*. The order of factual questions within these blocks was adapted during the course of the interviews to create a more fluent conversation. Generative questions complemented factual questions to allow interviewees to expand on some topics. We also included a final block of questions about their potential *use of tablets*. This revealed that tablet use is almost inexistent. Only three of our interviewees actually have a tablet but of small size and without a stylus. Only one of them uses a tablet to read scores. Three of them mentioned tablet use by musicians during performances. Four of them explicitly mentioned that handwritten input would be a great addition to either annotate or input notes.

We used contextual interview techniques [78] in order to situate answers in the context of interactions with concrete compositions. Three interviews were conducted at the composer's place of work, and 6 were conducted remotely with the possibility to turn on screen sharing when relevant. Interviews lasted between $\simeq$ 1 and $\simeq$ 3 hours. With the consent of interviewees, we took some pictures of their software sessions and paper scores to illustrate their comments. The interviews were recorded and transcribed.

Raw interview transcripts were then analyzed using two distinct methods, as illustrated in Figure 4.1. These complementary analyses were conducted with two objectives in mind:

- understand composers' individual *thought process* in order to capture valuable insights about their creative activities;
- understand composers' individual *work process* in order to identify both the positive and negative usability-related aspects of their music writing tools and workflow.

Figure 4.1 describes how these two analyses are articulated. The work process analysis consisted of a systematic application of the Cognitive Dimensions of Notations (CDN) framework [63] using the comprehensive set of CDNs listed in Table 4.2.

To perform this work process analysis, the first author meticulously reviewed all transcripts, identifying and categorizing specific comments that fall

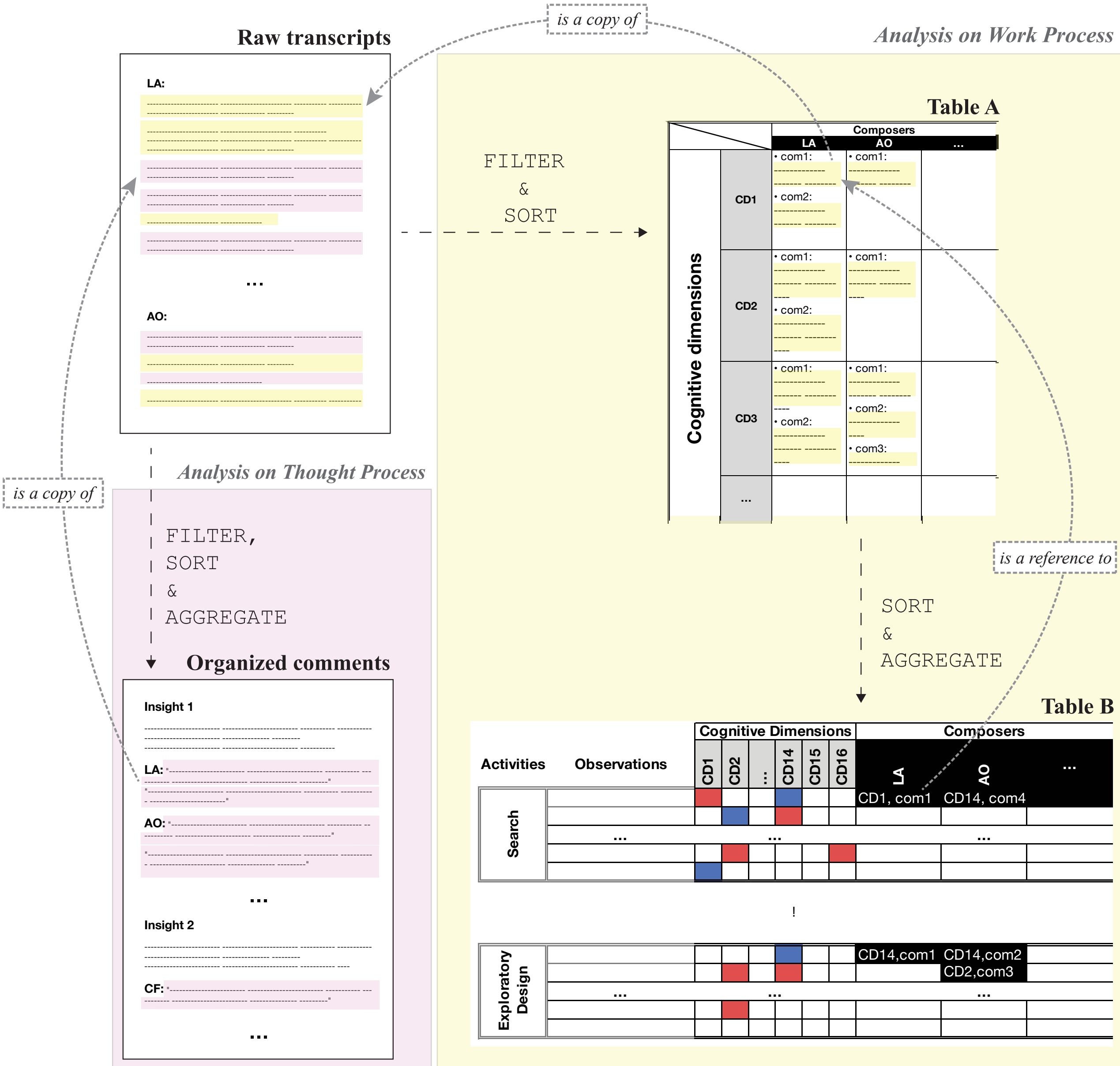


Figure 4.1: Methodology for analyzing raw interview transcripts.

under one or several cognitive dimensions (CD). This initial step yielded a table that indexes comments on the work process by $CD \times composer$ (Figure 4.1, Table A). These filtered comments were then further organized in a second table, wherein they were aggregated into higher-level observations and categorized based on specific activities as defined within the CDN framework (Figure 4.1, Table B). Observations were also color-coded, with positive aspects marked in blue and negative aspects in red. Table B provides explicit cross-references to the original raw comments, using the coordinate system established in Table A. We use this comprehensive analysis to present an overview of composers' work process in Section 4.2.2.

To ensure the accuracy and comprehensiveness of the CDN analysis, the second author conducted a review of the work carried out by the first author based on their own reading of the transcripts. This review led to some revisions regarding the classification under different CDNs and activities. Additionally, the second author collected comments that were not covered by the CDN analysis, comprising higher-level remarks from composers about their creative process. These comments were methodically sorted and aggregated into high-level insights, offering a comprehensive overview of composers' creative intentions and practices, which we report in Section 4.2.1.

By correlating observations about the work process with insights about the thought process, our overall analysis aimed to pinpoint areas where usability issues might impact composers' creative process.

4.2.1 Insights on Thought Process

In this section, we report on the analysis of the interviews along with aspects that emerged as crucial to understand diversity among the composers we interviewed. The primary objective of this analysis is to capture the composers' thought process as a set of insights. These insights will then serve as a reference to contextualize our analysis of their work process (Section 4.2.2) and to inform the proposal of design directions (Section 4.3). Our thought process analysis encompasses both aspects related to the composers' activities, such as: the various forms taken by the interplay of high-level and concrete concerns during the composition process; the importance of details; the assembly paradigm or the shaping of a specific individual; and aspects related to the properties of music, such as the role of time as experience or representation, and the mechanisms of repetitions and variations.

4.2.1.1 Interplay between Details and High-level Concerns

In this part, we highlight the importance of details to our interviewees, and the rich interplay of high-level and concrete low-level concerns during the composition process, which echo the different levels of musical abstraction in compositional activities observed by Roel [145]. As AO puts it, *"music is an art of details"*, with a constant interplay between abstract and concrete levels where music comes to life, emerges. Echoing Nash's observation about composers' need to *"access every detail of a piece, but also be able to get a sense of the 'big picture',"* [118] the important aspect to grasp is thus that there are generally micro and macro levels. Those need to be strongly linked together but have different properties, with much variation between composers. The form this interplay takes varies depending on the composer. For GF, there is *"a bit of dialectic between the macro and the micro,"* where *"the macro level would be the vision side, the micro level would be the purely auditory side."* For this composer, an abstract level view is required to clarify the state of his composition: *"sometimes you can't see, like I said, you've got your head in the sand [...] that's the macro side, that's it, the vision side."* But this is in contrast with MS, for whom reading at the concrete level of music notes helps making ideas clearer, since *"there are times when you're going to have to go through the score to see things more clearly."*

For some composers, the micro and macro levels are co-constructed by working on short musical segments. For instance, GF explains how the primary notation level (*e.g.*, rhythm) can also provide a macro level idea, as in *"Beethoven symphony number seven, the andante, there's precisely this macro rhythmic side that's «taa ta ta taa ta» and you can step back and see it higher and higher, it's always there [...] when you look at it from a macro perspective."* The emergence of a macro level can also come from an amplification of micro elements: *"It's a little world in itself that just needs to be enlarged, explored, amplified in as many ways as you want [...] Afterwards, if need be, like a painter, I like it to be stretched out a bit with water."* (GF). The emergence of form is compared by AO to the progressive sculpting of, *e.g.*, attack, resonance, orchestration. . . ; and by LA to polishing thanks to stronger ideas: *"once you start to give a form [...] you polish, you polish, you polish, because you have found more expressive ideas, better ideas, stronger ideas, ideas that correspond better to what you wanted to do."*

Details can be immediately-perceived data, pure matter (GF: *"ideas for themes come to us, and for us it's that and nothing else. It's obvious, it's pure matter, so to say"*) but also constraints, and tedious work in progress (*"GF: These places, we say to ourselves, are where there's the most craftsmanship, we're bound to say that it's not as good as art with a capital A and in fact these places are sometimes the ones we prefer too"*). What falls in the details category is also what is less essential, as in the case of JP who creates versions of a song with different levels of difficulty. To build an easier version: *"the idea is to make a slightly minimalist version of it, where we keep the essence of the song, the essential things and where you remove details or change specific aspects, for instance by keeping only a part of an arpeggio that is emblematic for the piece"* (Ständchen from

Schubert).

4.2.1.2 Experience of Time

Time, and the associated aspects of linearity and change, play an important role. As AO also says indeed, *"music is an art of time,"* which raises questions about the experience and sensation of time, but also about its representation.

For the composers we interviewed, the experience and sensation of time can consist either in putting oneself in the place of a listener *"there's bound to be listening, a human ear, a human temporality"* says MS, or in trying to live through a more linear composition process an experience equivalent to improvisation. In general, the composition process is not linear but rather highly iterative [38, 145], as several of the interviewees explained: *"It's never too linear"* says JP, *"I think that's a bit to be avoided"*. So, as highlighted by PG, *"it's not written as you go along, starting on the first note and ending on the last"*. However, AO explains the need *"to get back into the feeling of time and to get back into the improvised side of things [...] because composition isn't improvisation [...] In fact, right now I'm forcing myself to go back to something more linear, so as not to go too far into abstraction either."* GF, for his part, compares composing to a journey: *"I see [composition] as a process, as a journey [...] I really like initiatory things, I really like travelling and I think that each composition is a little journey and I think spontaneously I compose rather in a linear way."* This comes with *"happy accidents"*, as DR puts it: *"I like a bit of accidents, surprises and all that,"* and a relative reluctance to accept a pre-established structure: *"I like to give myself a certain amount of freedom when it comes to travelling."*

The sensation of time also encompasses the perception of change, which is why music often has to provide a sense of novelty, but within a familiar context. For this reason, repetition is important if the listener is to get familiar with the themes. To convey the perception of changes, the themes will be more or less transformed, as noted by Blackwell *et al.* [16] as well. As GF explains, there is a need to *"transpose or change melodic ideas, rhythmic ideas, or patterns."* It is important to think carefully about the number of repetitions, and about the type of those repetitions: *"I don't make music that's very repetitive in the sense of the motif, it's not the motifs that are repeated, it's more the sequences, the harmonic ranges, that sort of thing"* explains GC.

Regarding representation, our interviewees express a need to *"get an idea of what it would look like, for example in terms of time proportions [...] to work out the temporal relationships of different episodes in a composition"* (PG). For DR, *"finding your landmarks in time is easier with paper."* Other needs encompass landmarks in time, that can be provided either with bars, rhythms, pages (*"a page is like a rhythm"* as DR puts it) or musical events, and time durations that can be also provided with bars or with seconds. Knowing that *"such and such a bar lasts six seconds"* is important to GC, and for composers making movie music.

4.2.1.3 Shaping of a Specific Individual

The composers raised concerns about assembly, and how they work towards the shaping of a specific individual. They use all sorts of metaphors to describe how they see and manage the assembly of musical components to make a piece of music, from *"gap-fill text"* (GF) to *"jigsaw puzzle"* (MS) or *"sorter"* (DR). Their concerns relate to being able to put several ideas into dialogue, to mix them: *"because you're obsessed with the same things, it comes from you and it's material that will end up fitting together"* (GF) or superpose them: *"And here, I'm really working on the melodic motif that I'm going to associate with my rhythm, for example. So after a while, the two things will probably overlap. I'll superimpose them on the score, for example"* (LA). They also need to design transitions, because *"transitions are meant to unify the piece"*, as expressed by LA: *"It's like making a robot, you build the leg and the arm separately. And then you have to find a way for the leg and the arm to work on the same entity."* Composers indeed build on a more or less clear vision of the specific piece they are working on until they reach a state where something new, something that "works", comes to life, which they may compare to birth: *"But what often interests me is finding, let's say eight, twelve, sixteen bars, of an impactful melody that works with the harmony and that it all comes together as one. That's kind of my ideal, and we'll say that it's a birth, a raw material that already has a lot"* (GF). They are constantly on the search for something that will resemble the musical intention that leads them to the specific individual they are interested in: *"you have to find something that resembles the idea"* (LA), *"build it as good as you wanted it to be"* (CF), and *"make it real"* (GF). As we have seen above, a piece is made from a converging process involving low- and high-level elements that feed off each other: high-level, abstract elements (ideas or structure) are the ingredients of an architecture in which lower-level elements (notes, dynamics, rhythm, articulation) can take shape to form this musical individual. This process is well summarised by AO: *"In fact, there's matter and form. So it was Aristotle who distinguished between matter and form, if I'm not mistaken, and form is what gives it order. It's what makes it not just a collage of bits of music that we've put together without any vision behind it. Form is what makes it come alive, that it's not just matter, it actually has a shape"*.

4.2.2 Analysis of Work Process

We now report on our analysis of composers' work process, guided by the "Cognitive Dimensions of Notations" framework adapted to music notation [118]. We take the perspective of a notational *system*, *i.e.*, considering not only the notation itself but also the tools and environments to manipulate

Cognitive Dimension	Interpretation in the music notation context
Abstraction Management	"How can the notation be customised, adapted, or used beyond its intended use?"
Closeness of Mapping	"Does the notation match how you describe the music yourself?"
Error Proneness	"How easy is it to make annoying mistakes?"
Hard Mental Operations	"When writing music, are there difficult things to work out in your head?"
Hidden Dependencies	"How explicit are the relationships between related elements in the notation?"
Juxtaposability	"How easy is it to compare elements within the music?"
Knock-on Viscosity	"Is it easy to go back and make changes to an element?"
Learnability	"How easy is it to master the notation?"
Premature Commitment	"Do edits have to be performed in a prescribed order, requiring you to plan or think ahead?"
Progressive Evaluation	"How easy is it to stop and check your progress during editing?"
Provisionality	"Is it possible to sketch things out and play with ideas without being too precise about the exact result?"
Repetition Viscosity	"Is it easy to automatically propagate an action throughout the notation?"
Role Expressiveness	"Is it easy to see what each part is for, in the overall format of the notation?"
Secondary Notation	"How easy is it to make informal notes to capture ideas outside the formal rules of the notation?"
Synopsie	"Does the notation provide an understanding of the whole when you "stand back and look"?"
Visibility	"How easy is it to view and find elements or parts of the music during editing?"

Table 4.2: The list of cognitive dimensions, adapted from Nash [118] and Whitley & Blackwell [185], that we used in our analysis.

that notation as well [16]. As detailed in Section 4.2, we have systematically tagged participants' comments with the relevant cognitive dimensions and organized these annotated comments according to the different activities performed by composers. The five activities considered in our analysis are those listed in [17]: *transcription*, *incrementation*, *modification*, *exploratory design* and *search*.

It is worth emphasizing that nearly all composers engage in all five activities when using their music notation software. Throughout the remainder of this section, we maintain the same color-coding scheme employed in Figure 4.1-Table B to present our findings: blue for observations highlighting positive user experiences, and red for those indicating negative user experiences. Furthermore, we take care to provide context by referencing insights on thought process from Section 4.2.1 and from prior studies when relevant.

4.2.2.1 Transcription

Transcribing music requires composers to divide their attention between the source – whatever form it takes: original printed score, audio recording, live performance of an idea using an instrument – and the target – typically a score. It is a crucial activity across composers, who rely on multiple tools depending on the type of project and step in the process. These include their mind, paper, an instrument, an audio recorder, a computer. Ultimately, they need to produce a legible score (or set of scores) using the standard staff notation so that it can be shared with and played by musicians. In this section, we discuss the different kinds of transcriptions that may take place during the composition process.

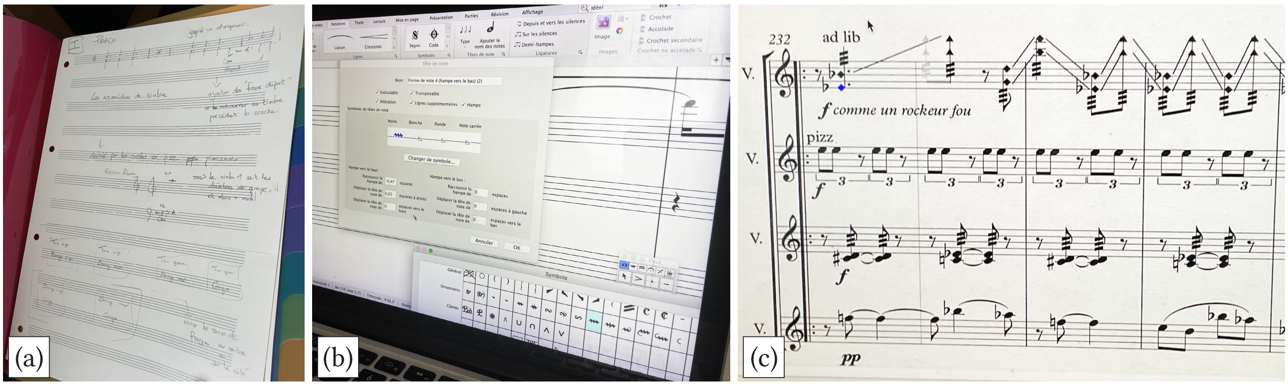


Figure 4.2: (a) Musical notations, textual annotations and sketches organized in a sorter. (b) Dialog window for the creation of a new abstraction. (c) Hidden bar in a passage that does not need the rhythmic frame of a measure.

From Immediacy to Computer

Transcribing musical ideas coming from the mind, whether writing them or playing them, is akin to capturing fleeting thoughts. As note-taking, the transcription process needs to be as smooth as possible so as not to impede the flow of ideas [148]. For instance, CF, having established a basis of raw ideas in her mind and on paper, concretizes them by entering them on her computer: *"I use [the computer] more like a typewriter."*

However, inspiration can occur in situations where neither paper nor computer are within reach. As DR explained: *"Ideas often come like that, when you are not at your desk [...] you take a step back and it comes."* Seeking a ***linear flow*** (see Section 4.2.1.2), many composers find their ideas while playing their instrument. They do not want to interrupt their flow to write their musical ideas down. They might record themselves using an audio recorder, adding yet another source of information that will later have to be transcribed since audio input is rarely integrated in music notation software. Because they are created independently, these recordings can contain discrepancies with the rest of the music piece when several instruments are intermingled (high level of hidden dependencies), resulting in ***assembly challenges*** (see Section 4.2.1.3).

From Paper to Computer

The score that gets passed down to musicians must be computer-written for readability purposes. In order to avoid spending too much time on transcription, most interviewees base their work on computer and use paper as a secondary tool. Nonetheless, paper remains an efficient way to capture fleeting ideas [144] and to engage in unbounded creative processes, both because it is free of the many constrains imposed by software and hardware, and because it is the way composers learn to write music. They can put down exactly what they have in mind, either ***detailed notes or high-level ideas*** (see Section 4.2.1.1) to build their ***specific individual*** (see Section 4.2.1.3) and worry about the rest later. Paper is *"a private medium [...] tolerant of errors"* that composers *"are free to misuse"* [118]. Writings can thus be musical notations (symbols), but textual annotations or freeform sketches as well (Figure 4.2-a).

When transcribing on the computer, composers can choose between learning the numerous keyboard shortcuts (low closeness of mapping and learnability), navigating through the icons and menus (high knock-on viscosity due to the low visibility of the overloaded menus), playing the melody on a midi keyboard (high error proneness but high closeness of mapping for piano players), or a combination of those. Most interviewees seemed satisfied with how they are transcribing basic musical notation. But many complained that some symbols are not supported by their software and require tremendous effort to add (low closeness of mapping and abstraction management), sometimes requiring the creation of a new symbol such as an accent or note head (Figure 4.2-b).

Textual annotations are not necessarily destined to be transcribed on computer scores. For instance, personal comments or research notes are usually made on paper, because music notation software are centered on the score and have limited support for such side-notes (poor support for secondary notation). Similarly, sketches such as structural plans are made on paper because they are not supported by the software. Even if they were, drawing them with a mouse or trackpad would be particularly impractical. Those handwritten notations, when bounded to stay on paper, force composers to divide their attention between mediums. More importantly they force them to be perfectly organized to be able to access the information when needed (as AO who writes in notebooks or DR who uses sorters, see Figure 4.2-a), whether it be during the composition process or years later.

From Computer to Computer

Arrangers often rely on PDF files or have several instances of the application opened simultaneously to transcribe parts of musical notations. This can bring visibility problems, as JP stated: *"I have the PDF with the original score on one half of the screen, another PDF with a piano arrangement which I draw ideas from, and Finale roughly above, so I switch from one window to another."* LA also complained about the impossibility to put two scores side-by-side (low juxtaposability), leading him to make constant back-and-forths between files and remembering musical elements to transcribe (hard mental operations) instead of making a visual copy.

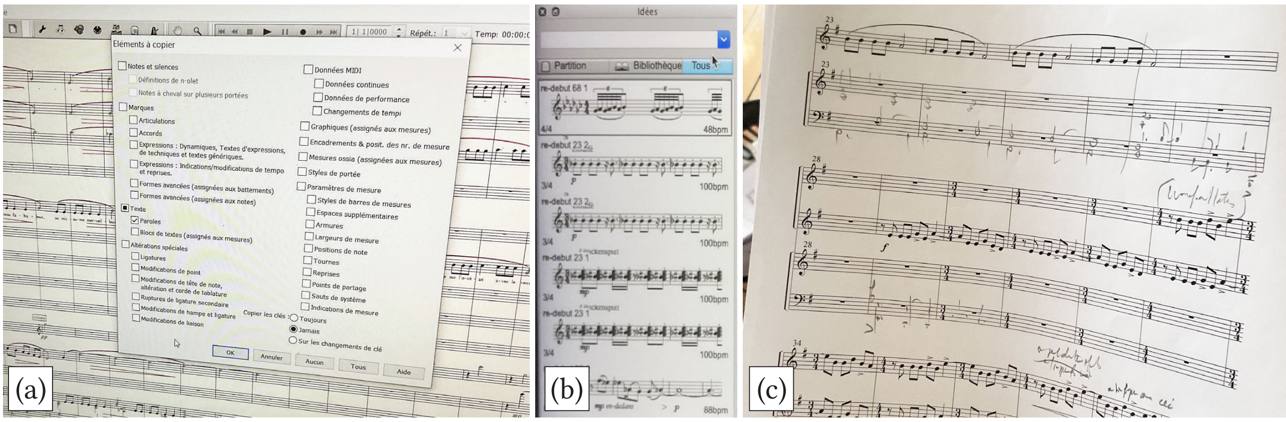


Figure 4.3: (a) Dialog box used to specify properties to copy. (b) Vertical menu to store ideas aside from the score. (c) Written orchestration ideas on a printed score.

From Computer to Paper

The last type of transcription is from the score to the end product(s): when written for several instruments, musicians generally only need their own part. In this case composers must extract the respective staves, producing new scores that can be distributed to - and usually printed by - the musicians. Music notation software handle these extractions, but while composers do not have to rewrite the score's contents, they have to adjust the layout because measures and spacing between staves get jumbled in the process (high level of hidden dependencies). Moreover, when the music piece is split between several files, composers have to repeat these extractions, increasing repetition viscosity, as corroborated by CF : *"I do this file by file, it takes more time."*

4.2.2.2 Incrementation

Adding small bits of information to the music notation differs from transcription in that it does not require composers to divide their attention. They can remain primarily focused on the score. Moreover, incrementation not only involves adding content – such as notes, symbols or text – but modifying or erasing content as well. In this section, we focus on the context in which composers perform this activity.

The Sacrifice of Flexibility for Aesthetics

A score made with music notation software is both beautiful and well-structured. Resembling an engraved score throughout the work process (providing high visibility and progressive evaluation), it can help composers with some tasks. But it also impedes other tasks. For instance, AO writes a lot of annotations for himself on the score and needs to get rid of them when sharing the final product. He has trouble identifying these annotations because they look too similar to the primary notation (poor support for secondary notation). Another major problem of working directly on the final product and not on a personal draft is the negative impact on creativity, because the rigid framework that comes with this appealing notation imposes rules on composers that sometimes prefer more flexibility (low abstraction management). As GC said: *"It's configured in every way, so to be able to unconfigure something is not easy, you have to really dig into the depths of the software, and the user is not at all creative with the tool. He can't create his own tool, he can't develop, except by being super strong. But otherwise it's true that we endure a lot and in fact we can do things but it's always tricky, tinkering."* MS wished for an interface closer to his mental model, that would not hinder his workflow: *"I would just like to write it in a more graphic way"* (low closeness of mapping).

It is indeed difficult to add notations because composers need to deal with issues of form and position, whether it be within measures (*e.g.*, elements that shift when you add new ones) or around them (*e.g.*, additional lines, ornaments) (high knock-on viscosity). But the visual issues are symptomatic of deeper, more logical ones. For instance, CF commented about being unable to step outside of the rhythmic framework: *"The starting point is the measure. But when we don't have measures [in our music], that's a lot more complicated because we are still obliged to pretend we use measures, to delete them afterwards, to ensure that they don't appear anymore and that the spacing is still consistent"* (Figure 4.2-c). LA shares this concern: *"In these software, the rhythm is hyper framed. If you make music in 4/4, everything has to be in 4/4 perfectly. It can't overflow. You are forced to solve rhythm problems where sometimes you just want to put your idea. And sometimes it goes beyond the measure."* Such problems, that do not exist on paper, force composers to look for answers in the software's documentation or on the Internet, leading them to think twice before adding a complex notation (low provisionality) or to postpone their action, as GF: *"There are places that we deliberately leave under construction, we tell ourselves that we will come back to them later"* (high premature commitment). Because phrasing a practical problem to get help from online forums is troublesome, some composers give up and avoid using music notation software to their full potential. On top of that, elements such as nuances or fingerings are also rather added towards the end, in order to avoid having to repeat this task in case of more fundamental changes to, *e.g.*, the rhythm or melody.

The role of Copy and Paste

Once composers have written something they like, they do not write it again. They rely instead on the copy & paste function, especially when they are working with ***repetition and variations*** (see Section 4.2.1.2). The ability to instantly reproduce elements that have been hard to input is one of the strengths of music notation software. It represents a huge time-saver and makes incrementation easier by handling the heavy lifting in place of composers (*e.g.*, pitch and rhythms of numerous notes, or weird abstractions), making up for the time lost inputting the notation initially. As GF explains, there is a point where copy & paste is much more frequent than manual input of new symbols. But while it is a central function, it could be more efficient. Copy & paste comes with a lot of hidden dependencies, because the software does not know which part of the notation the composer is interested in. It needs to be specified, if the composer only wants to copy the rhythmic aspect of a motif and paste it in a different harmonic context, or the other way around in case of different time signatures. Most interviewees struggle with it, and only CF mentioned an advanced, arguably bloated dialog box that enables customizing the copy & paste function (Figure 4.3-a).

Besides individual passages, copy & paste is also of use in the context of more global actions, when the score contains many repetitions, or features *"a theme returning in a modified form"* [16]. This quickly becomes tedious, as composers must *i)* ensure that they are copying the right elements in order to avoid bad surprises when pasting; *ii)* search for relevant occurrences through the notation [16] and paste in the right place; *iii)* perform modifications on the elements in every passage to edit (high repetition viscosity) – in addition to layout problems discussed in the next section. Some composers wish they could apply grouped edits on motifs of interest.

4.2.2.3 Modification

The reorganization and restructuring of existing notation focuses on its layout and implies that no new content is added to the score. Composers often have to modify the layout of individual elements or entire measures to improve the readability of the score. But multiple dependencies, between elements and measures and between elements themselves, make this difficult. We detail the three levels at which we observed such dependencies.

Elements within a Measure

Notes and accidentals are usually bound to a measure. Music notation software need to anticipate what composers are aiming for – as opposed to paper where elements can be written directly as intended without interference. The software often makes wrong guesses (resulting in high error proneness), misplacing elements (low closeness of mapping) which then require additional operations to correct. LA complained about this gulf of execution: *"If you want something really specific, you cannot get into it and you always stay at the surface of it. [The software] might place elements in an intelligent way, you don't want them to be placed like that. It makes an average between what it thinks is best and your* *intention, so you waste a lot of time trying to adjust things progressively."* Actions like shifting notes, flipping stems, or adjusting accidentals can be tedious to perform in the confined space of a measure (high knock-on viscosity) because of the structured editor's rigidity, as discussed earlier. As MS said: *"You tinker with it to make it fit into a frame."* When the notation is dense, composers will often start over rather than embark on a complex edit sequence.

Elements within the Page

On the contrary text, and most symbols, are not bound to a measure and can be moved around freely on the page. While this can be an asset when ***fine-tuning the layout*** (see Section 4.2.1.1), it can also create problems, impeding the ***restructuring and assembly of musical components*** (see Section 4.2.1.3). LA, for instance, who *"would like to drag things around"* and *"move measures, but refrain from copy-and-pasting"* has currently no other choice, and because texts and symbols like dynamics and ornaments are still conceptually bound to the elements of a measure, they need to be selected separately.

Measures within the Page

The layout of measures constitutes a major source of friction, as expressed almost unanimously by the interviewees. Layout decisions can depend on diverse constraints, like aiming to have a fixed number of measures by staff, the need to fit the score in a given amount of pages, or the effort to match layout interruptions with the flow of musical phrases – avoiding a page break in the middle of a phrase, for instance. Whereas a text editor can go as far as hyphenating a word to optimize layout, the modern staff notation forbids splitting measures. Composers thus resize measures at the risk of triggering a snowball effect on the measures that follow due to their high hidden dependencies, leading to much hard mental operations. As AO explains: *"It has to be somewhat standardized. You have to ensure that all the notes have approximately the same gap, but at the same time, that all the pages are almost full, that everything is filled, that there is no empty space, unused space. Yeah, that might take a little thought."* Some composers do work on the layout of measures during the writing process for the purpose of progressive evaluation, to get a sense of the final structure. But as observed by Bennett [14] most music writers delay the effort, possibly until the end (too much premature commitment), preferring to work in a fickler context in order to avoid the potential waste of tedious layout efforts.

4.2.2.4 Exploratory Design

It is common practice for composers to play around with new ideas on their instrument and only put them in writing after they are satisfied. However, going through writing for exploratory design without being sure of what will result allows to see and understand the music mechanisms, favoring progressive evaluation. As illustrated by MS: *"When there are several voices, it allows you to see how they can fit together so that it can work [...] Seeing the notes in writing makes it clearer in the head. Otherwise it's not intellectualized, and it stays a bit vague."* We discuss several strategies to support exploratory design, each adapted to its situation.

On the Final Score

Exploring ideas directly on the final score is risky, as it can adversely impact work done earlier, be it content- or layout-wise. Composers thus usually restrict their in-context modifications to small edits only, with manageable hidden dependencies over content and close-to-none effect on layout. When faced with more ambitious modifications, another strategy described by GC consists of copying a passage and pasting the duplicate right next to it, working on that duplicate and thus staying in-context. This gives music writers high juxtaposability power, allowing them to try things without altering the original material, and changing the layout only temporarily (better provisionality). However, a more popular strategy is to copy and paste the passage at the end of the score, in measures left blank to this effect. Composers then work in this isolated zone, akin to a sandbox, *i.e.*, safe, without hidden dependencies, but at the expense of juxtaposability and visibility. Committing modifications then requires effort to ***assemble the musical components*** and make them ***work together*** (see Section 4.2.1.3).

When working on the final score, another issue to consider is the lack of versioning capabilities [65] to keep track of changes. Even if editors have an undo feature, edits quickly become definitive. As GF said: *"There is a lot of choosing. And sometimes the choices are painful because there are things we like and would like to keep, but we must not hesitate to erase."*

Away from the Final Score

In some cases, an idea turns out to be a better fit to another piece. Or composers simply want to explore it later. Some software feature a component where to store such ideas, akin to an elaborate clipboard that can be accessed and browsed (Figure 4.3-b) from any score. Finding a particular idea in the stack and working on it can be challenging, however. Most interviewees also explained switching between several files in order to explore novel ideas. For instance, some of them maintain a draft version and an engraved version, or a succession of chronological versions of the same score. Composers thus reassure themselves with the existence of a static version before embarking on global modifications (*e.g.*, changing the tonality) that could disturb the whole piece in ways difficult to predict. Some composers rather print their current score and scribble on it with a pencil, to regain *"the expressive freedom of pencil and pen marks"* [118] they need to explore ideas (high closeness of mapping). CF does so to try out orchestration bits, under the lead parts that are software-written (Figure 4.3-c). DR explores distinct aspects of his music on paper (*e.g.*, rhythmic, harmonic or contextual ones) and explains that it also entails low premature commitment: *"I can always note new ideas, new concepts, there is no precise order."*

4.2.2.5 Search

Blackwell *et al.* [16] identified the search for occurrences of a theme or motif as one of the generic activities performed by composers. More specifically, our observations reveal that composers search for information within the score for two main reasons: when looking for an element – typically to edit it – and when reviewing their work, checking for possible mistakes, making sure everything is "as it should be" – ***shaping a specific individual*** (see Section 4.2.1.3). We identify search strategies available to the composers and discuss the importance of the chosen perspective.

Using Absolute Positions

Music writers can find an element based on its previously-known location. For instance, bar or page numbers are useful to find a location in relation to another source, acting as a coordinate system. CF, who writes on paper before working with software, explained: *"The paper version does not have the same layout, so visual cues change. But there are bar numbers that do not change."* JP, who writes arrangements using software together with the original score displayed as a PDF, added: *"Sometimes there are repeated elements, so to ensure that I am in the right place, I use the bar numbers a lot. [...] This is mostly for comparison purposes"* (high visibility). Although they are easy to jump to, bar numbers are not always there in the first place, limiting juxtaposability: on a handwritten score composers have to write every number themselves, and on an original score numbers are usually shown for the first measure of a line only.

Using Relative Positions

Comparison tasks aside, numbers are of little use to find an element in a score, as they have no meaning musically-speaking. Instead, composers use their ***knowledge of the form*** (see Section 4.2.1.3) and ***temporal grasp*** (see Section 4.2.1.2) of the score. MS explained: *"I know that [what I am looking for] is towards the end, or the beginning, or the middle of the score."* PG relies on relative positions: *"It is before or after another musical event, theme or passage that I remember."* Although this type of search works fine with short scores, it quickly becomes tedious with longer ones, as LA stated: *"Finding a precise element across twelve pages begins to feel like finding a quote in a book."* To reduce the search scope, composers commonly resort to custom visual cues such as headers, comments and color codes that catch the eye and that can be quickly identified from an overview of the score. These spare the composer from having to actually read the notation in detail, acting as bookmarks that support role expressiveness.

Using Musical Properties

Some interviewees expressed the need to find elements based on their properties rather than by scanning and navigating the score, essentially calling for better abstraction management at both the ***micro or macro levels*** (see Section 4.2.1.1). For instance, GC wished he had a way to quickly inspect specific details: *"I know what is missing, what would be great: an internal search engine. If I am searching for a succession of notes and/or rhythms, I would like to find it like when I do "Ctrl+F" in a text editor."* It is also the case for AO, who needs to check that every time he writes *"pizz"*, he also writes *"arco"* somewhere in the following measures. LA expressed related high-level concerns: *"I would like to search for precise things in a score by their theoretical designation or by describing their role in the piece. For instance, I may need to find a "transition" or a "conclusion"."* This corroborates what GC said about studying his own music, concerned about the ***temporal experience*** (see Section 4.2.1.2) of the listener: *"You are not always aware of everything. From a formal point of view, you may ask yourself: "Have I not over-used this motif ?". [...] What is hard to realize when you are composing, is the effect it has on someone who listens for the first time. [As composers], we don't have this initial listening experience and need to check if an element is repeated too many times, or not enough to actually be remembered."*

Using Ears and Fingers.

Composers often rely on the audio playback function to check their score for mistakes. Playback, even with mediocre sound quality or with the wrong instruments playing, lets them check the notation in ***detail*** (see Section 4.2.1.1) and reveal hidden errors of pitch and duration. It is particularly useful among composers who can neither perform what is written with their instrument, nor imagine the result. Even for those who can, it hints that the musical notation is fairly opaque, thus bearing high error-proneness. As CF said: *"The ear is better than the eye."* Progressive evaluation seems to be best supported by enabling composers to listen and read simultaneously in order to check that the sound matches the notation.

However, when composers assess the musicality of their work, the audio playback function usually falls short: elements around the staff, like dynamics and ornaments, although being an integral part of the notation, are not always recognized as such. In addition, digitally-simulated instruments fail to sound realistic and have limited expressivity (low closeness of mapping), forcing composers to install additional sound libraries. Only then can they take a step back from their work and properly gauge if the musical elements are well-balanced, ***together*** (see Section 4.2.1.3) and ***over time*** (see Section 4.2.1.2).

Another piece of information that can be found neither by reading the notation nor by listening to the piece is the *playability* of the score. Fingerings, complex chords and hazardous rhythms must be tested on a real instrument, because these are interpretation-related problems. Many composers, including GF, exchange with performers to iterate on these details: *"We show the score to the violinist or the pianist, and we are told: «It will take two weeks to work on just these two measures, it's not consistent in terms of difficulty.» There must be a certain coherence in the playability so that the performer can express himself well. The music must fall under the fingers."* These remarks are usually noted on the spot using paper. The composer then has to find the corresponding places to modify in the score, leading back to the first issue discussed in this section.

Using a Specific Perspective

Regardless of what composers are searching for, how the score is displayed heavily influences their ability to navigate easily. Being able to switch representation depending on the activity is a key feature. For instance, when the score is written for a solo instrument, the portrait orientation is preferred. As AO explained: *"I am able to see almost half of my piece on one page, it gives me a more **global vision**"* (see Section 4.2.1.3) (high synopsie). When the score is written for several instruments, the landscape orientation is sometimes a better option, reducing the number of line breaks (high visibility). These two types of display are both paginated, bringing ***temporal structure*** (see Section 4.2.1.2) that can help music writers like DR: *"Pages allow me to compartmentalize. They set the pace. I know that there is roughly the same duration on every page. […] They are like the white bands on the road that show you how far is the car in front of you because without them, it would be much harder to have a notion of space and* *time"* (high closeness of mapping). However, this structure also pushes them to deal with layout challenges early on, shifting the focus away from creativity (high premature commitment). Panoramic orientation removes pagination and shows the score as a continuous flow of measures, reducing premature commitment. CF explained: *"It depends on the state of the work, and it is very variable. Usually, it is easier to start like this without worrying about the layout, and then work on the layout, which is tedious."* Composers can compensate for the lack of systematic structure with their own abstractions like the headers discussed earlier. AO, writing a piece of music for a movie, comments: *"The score has five parts that correspond to five scenes, and I see them visually, because I defined them with numbers."* This way, good abstraction management increases visibility.

The longer the score gets and the more instruments a piece involves, the more important structural choices become. Composers often split their score in several files to work on distinct parts in any order, reducing premature commitment and making it easier to find a specific element (high visibility). Another way of increasing the visibility is to optimize the space by hiding non-essential staves (for instance those with no notation) during the writing process, displaying them only in the end.

Taking a step back in order to get a sense of the global flow allows composers to find elements or spot mistakes easily, provided that they can still see the details: when the score is several pages long, unzooming quickly makes the notation illegible (low synopsie). This is why many composers use a wide aspect-ratio monitor, several monitors or even a printed version of their score that they can hang on the wall or spread on the floor, increasing visibility, sometimes at the cost of additional transcription tasks.

4.3 Design Opportunities

At a high-level, our analyses suggest that music writing software impose too many constraints when editing the primary notation, in line with observations made by Nash [118]. While a couple of composers acknowledge the utility of strongly-structuring elements such as measures or pages to guide their composition process, the majority of them clearly express that, although they want structured outputs that they can share with other musicians, music writing applications frequently interfere with their creative workflow. This often leads them to find workarounds to cheat the software or to resort to more adaptable mediums such as paper and audio recorders.

Opportunities to address these issues are both at the notation and at the environment levels [64]. Notation level design manoeuvers include, *e.g.*, reducing viscosity by providing relevant abstractions, or increasing clarity by enabling secondary notations. At the environment level, as discussed in Chapter 2, the integration of interactive surfaces supporting digital pen input in the music writing process offers a promising avenue for increasing flexibility without sacrificing structure by *"simplif[ying] the alteration, erasure and overwriting of notes and passages"* [118]. An obvious advantage of digital pens lies in the easy creation of freeform annotations above a structured digital representation. As observed at multiple occasions during our interviews, music composers have their mental representations towards the ***specific individual*** (see Section 4.2.1.3) they have in mind, partially implemented in their score but not entirely in the primary notation. Freeform annotations can reproduce the experience they have with paper where they can circle, add symbols and notes to capture their mental representation of the score and/or convey intentions to the musicians who will perform their piece. However, we believe that the advantages of digital pens go much further than supporting the easy and intuitive input of annotations [189]. In particular they have the potential to address issues that are related to the rigidity of the structure. This section discusses opportunities that seem worth investigating as part of the design of future music notation software. All of these opportunities revolve around the idea of temporarily breaking the structure to better fit composers' thought process. We envision breaking away from the structure along three possible directions, detailed in the remainder of this section:

- breaking down musical elements: enabling manipulation of score elements at a finer granularity than the primary notation permits;
- breaking the score's homogeneity: allowing composers to capture their ideas using other notations and media *within* the score;
- breaking the score's linear structure: offering composers the freedom to arrange scores and musical fragments spatially, adapting the layout to the specific task at hand.

4.3.1 Breaking Down Musical Elements

Our analyses revealed that, in their building of a rich ***experience happening within time*** (see Section 4.2.1.2), composers dedicate much attention to repetitions in their work, inserting recurring patterns but customizing some or all occurrences of those patterns so that repetitions become variations rather than mere duplicates. Composers work with various aspects of patterns: rhythm, pitch, fingering, nuances, ornaments. They explained heavily relying on copy-paste to facilitate such pattern-centric work but, because copy-paste is monolithic, spending a lot of time to add, remove and adjust elements within the pasted fragment, as we detail in our analysis of their incrementation activities. Furthermore, our examination of their search activities revealed that music notation software lacks effective means to identify musical segments based on specific musical or notation properties.

The precision and flexibility of the digital pen could empower composers when working on patterns. The pen can *precisely point at and delineate* graphical elements. Current music software only allow monolithic selections, but with this type of precise selection input device composers could tell the system what specific properties they are interested in. This could for instance be through direct selection by, *e.g.*, circling them or making a freeform lasso selection on a specific element in order to restrict the selection scope to only elements of the same type. For instance, they could initiate a selection on a beam (resp. a note head) to copy only the rhythm (resp. the pitch) within the delineated fragment. Or they could select and copy only the fingerings to be repeated throughout the score – an operation that was described as particularly tedious in our interviews. Beyond the direct selection of elements, the pen could leverage users' ability to *write and draw elements by hand* beyond input of the primary notation to search patterns in a score and thus facilitate checks and edits across repetitions. Interviewed composers reported the lack of support to search patterns according to diversified musical properties. A pen-operated search box could address this limitation. Composers would be able to express the pattern they seek by drawing note heads, beams, stems, or combinations thereof, specifying a query across their score to identify fragments that match their desired pattern, as illustrated in mock-up (a) in Figure 4.4. Based on what gets highlighted in the score, they could draw or erase some ink marks in the search box to further constrain or relax their query.

4.3.2 Breaking the Score's Homogeneity

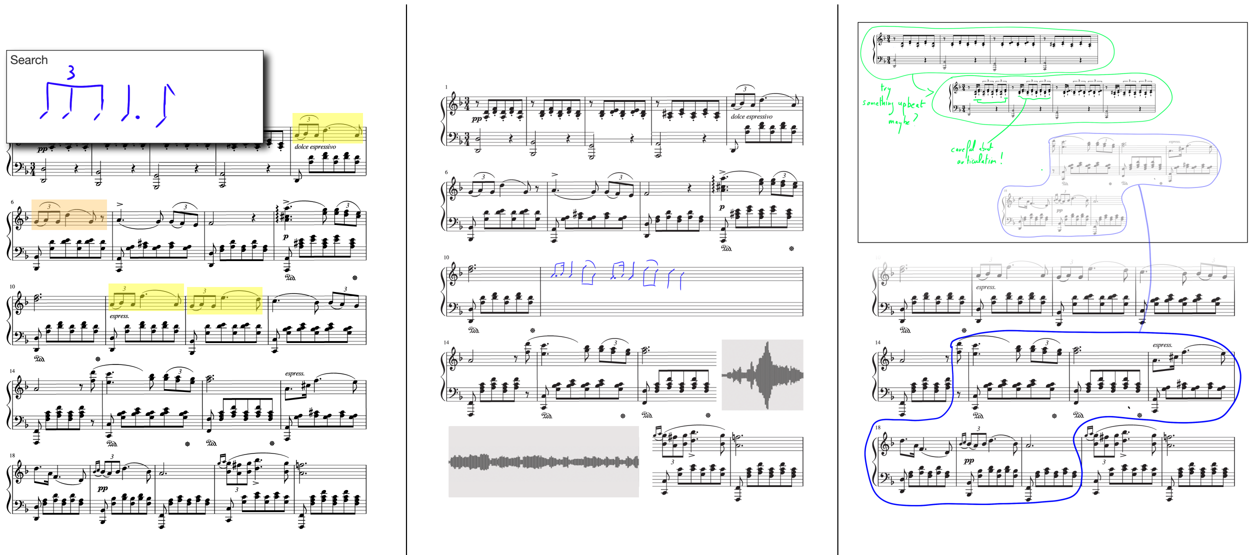
Our interviews revealed a potential mismatch that can occur when the temporal flow for writing music in software is not aligned with the natural flow of composers' creative thoughts. This creative flow is different depending on the composer's thought level: at a low ***level of details*** (see Section 4.2.1.1), it is rather temporal and auditory, while at a ***high level***, it is more about spatial organization, considering parts, structure, and relationships, which are predominantly visual in nature. Our analysis of work process revealed that composers often pause their software-based work to turn to alternative mediums such as paper to capture fleeting ideas in the form of notes and sketches, or an audio recorder to capture musical sequences played directly on their instruments. Additionally, they are often constrained by the software's rigidity in terms of form and layout, when they would rather have the flexibility to leave certain sections in draft form for later refinement. In essence, composers' thoughts cannot always be expressed in the primary notation supported by the software, leading to the usability issues detailed across the Transcription,

(a) (b) (c)

Figure 4.4: Mock-up illustrations of design opportunities: (a) using the pen to sketch a rhythmic pattern to be searched in the score, differentiating between perfect (yellow) and partial (orange) matches; (b) inserting heterogeneous content in-context: handwritten symbols that do not get automatically beautified, and a playable audio recording represented using its spectrum; (c) selecting (blue stroke) and copying measures to a canvas where passages remain editable but can be laid out freely, ignoring layout rules imposed by the staff notation.

Incrementation, and Modification activities in Section 4.2.2.

Current music notation software expose scores as one homogeneous sequence of staves that cannot be broken and mixed with other content. To address the above issue, the score editing environment could be made more flexible, breaking the homogeneity of the score to support the insertion of different types of contents and media both within and between staves. These elements would not persist on the score. They would exist only temporarily, until they get transformed into actual music notation. Their purpose would rather be to enable composers to insert ideas in-context, regardless of the way they were captured. Starting with a relatively simple case, recent work on supporting in-context annotations on pen-based devices can inspire interaction techniques for music notation software. For instance, SpaceInk [150] enables users to insert handwritten notes between words, lines, or paragraphs in a structured text document by locally reflowing its contents. RichReview [199] also explores the concept of in-context annotations with a rich variety of media, including audio annotations. Such systems preserve the spatial structure and linear flow of documents without being overly constrained by the original document's primary notation. Applying a similar approach to music notation software, composers could for instance insert empty space between existing measures on a staff to jot down a new idea (3rd grand staff in Figure 4.4-b). They could also insert ideas captured as audio recordings, represented using their audio spectrum (Figure 4.4-b across the 4th and 5th great staves).[2] In both cases, the new idea is placed in-context *effortlessly*, as it does not have to be converted to the primary notation immediately. This can be postponed to a later time, causing less interruptions to the composer's train of thoughts.

Whichever notation we consider, offering flexibility between *interpreted* input (digitally-enhanced musical symbols) and *non-interpreted* input (handwritten symbols) seems particularly important. Non-interpreted input lets composers capture fleeting thoughts without disrupting the creative process. Composers may choose to leave a region in the form of annotations or as elements from the primary notation but with relaxed constraints (*e.g.*, without mesure bars) and refine it later (3rd grand staff in Figure 4.4-b), eventually asking the system to interpret when it is stable enough. This is opposite to what music notation software such as StaffPad does, as it recognizes hand input notation greedily – see Section 2.3. It seems particularly important that composers keep control over when a passage gets interpreted to avoid the system taking premature actions such as adjusting the layout while they are still in a transitional phase of modifications (*e.g.*, when filtering their score for a specific instrument). Pen-based systems such as MusInk [179], WritLarge [195] and ActiveInk [148], which give users much freedom regarding when to interpret pen input, can inspire the design of such interactions.

4.3.3 Breaking the Score's Linear Structure

Linearity is a fundamental aspect of a musical piece. Composers assess their scores with respect to temporal evolution and temporal constraints. However, the creative process itself is often non-linear. It can even extend beyond a single composition. In particular, in their ***complex and iterative assembly process*** (see Section 4.2.1.3), composers frequently revisit specific sections within their scores to explore alternative versions. They also sometimes draw inspiration from their own previous works, or from the works of others. Unfortunately, music notation software poorly support tasks that involve sections scattered within a score or across scores. Additionally, when revising a section at a ***detailed level*** (see Section 4.2.1.1), composers often copy the section and paste it right after the original to work on that copy (potentially disrupting the layout). Or they copy it towards the end of the score, potentially losing valuable context. Some composers even prefer creating several independent

[2]While some systems like StaffPad feature the possibility to add an audio track, this track runs parallel to, and independently from, the staves. It rather serves to accompany playback with foreign audio.

versions of their score. This is due to the rigidity of the score's linear structure. A well-designed pen-based system could allow temporarily breaking this rigidity to facilitate the activities that this rigidity adversely impacts, as identified in our analysis of the work process: Transcription, Incrementation, Exploratory Design.

One particular advantage of pen-based systems is their ability to replicate the experience of working on a blank sheet of (unstructured) paper. In digital systems, this concept can take the form of an infinite canvas where users can insert various elements and arrange them freely [194, 148, 146]. Designing such a *canvas mode* would be a valuable addition to music composition software that could support much more operations than the Idea Clipboard depicted in Figure 4.3-b, which is essentially a list of archived passages that can be pasted on a score. With a more flexible canvas, composers could organize and *"retrieve [collected ideas] based on diversified cues"* [151]. Figure 4.4-c illustrates how such a canvas could be used as a storage space as well as a sandbox to experiment with fragments safely. Those fragments could be arranged freely to support spatial grouping and indexing, and to support tasks that involve their comparison side-by-side. This flexibility would for instance help address the challenges composers encounter when transcribing content from one score to another. It would also facilitate exploring different versions of the same passage. With a canvas mode, composers could work on different versions concurrently, add annotations as needed, and then preview and transfer some of them back to the original score, effectively committing their changes. By combining an unstructured and infinite canvas with annotation capabilities, composers could keep a record of their creative process, that would thus favor reflection when needed. Transitions between these views can be made seamless in pen and touch input systems, thanks to, *e.g.*, command marks [6] or menus designed for quick access to frequently used commands [130].

4.3.4 Discussion

The proposals presented above give a glimpse of how the affordances of interactive surfaces could help address composers' contradicting needs for structure and flexibility. They were chosen as representative examples but they by no means represent a complete solution. Much remains to be done and developing means to better support the creative process in music notation software will require careful interaction design. As evidenced by the literature on pen and touch interaction, multiple challenges need to be overcome. New and existing features must form a coherent whole. Essential features need to be exposed primarily as direct manipulations seamlessly integrated in terms of input. In addition, the legacy of designs implemented by existing

Interviewee	Initials	Activities(*)	Main Style	Experience (y)	Current Software	Main Writing Tool(s)
Lucius Arkmann	LA	C, A, E, T	Modern classical	> 15	Musescore	Laptop or paper
Caroline Itier	CI	C, A, T	Jazz	> 15	Finale	Laptop or paper
Bruno Fages	BF	C, A, T	Jazz	> 30	Sibelius	Laptop or paper
ANONYMOUS	AN	C, T	Modern classical	$\simeq$ 20	Sibelius	Laptop and paper
Alexandre Olech	AO	C, A, E	Modern classical	> 5	Sibelius, Dorico	Desktop
Yves Torchinsky	YT	C, A, T	Jazz, Rock	> 40	Sibelius	Laptop and paper
Olivier Sabatier	OS	C, A, T	Modern classical	> 20	Sibelius	Laptop and paper
Dominique Pifarély	DP	C	Jazz	> 45	Sibelius	Desktop, laptop and paper

Table 4.3: Profile of the eight composers we worked with. *Activities: C=Composition; A=Arrangement; E=Engraving for another composer; T=Teaching. All of them participated in the final interviews and two of them also participated in the design workshop (we refer to them as C_1 and C_2 for anonymity concerns).

software, that composers have invested significant time to master, cannot be ignored.

An additional challenge lies in the integration of interactive surfaces into composers' work environment. Rather than an immediate and full replacement for a desktop workstation setup, we envision interactive surfaces serving in two situations: 1) as the device of choice when away from the main workstation (mobile context of use); and 2) as a complementary tool alongside more traditional mediums such as paper and desktop computers [16]. An effective interaction design will likely position them as a preferred option over paper in certain scenarios or over computers in others. We even anticipate that, for some composers, interactive surfaces may replace the need for either paper or computers entirely. Ultimately, interactive surfaces have the potential to enable a more radical change in composers' work environment where both paper and computers would be replaced by a single, comprehensive, and efficient medium.

4.4 EuterPen: A Music Notation Program Prototype

Building on the guidelines of Section 4.3, we developed EuterPen, a music notation program prototype Figure 4.5 designed specifically for interactive surfaces. The development of EuterPen followed an iterative design process that consisted of four main phases: a first design and prototyping phase informed by the guidelines; a workshop involving two professional composers; a second design and prototyping phase informed by this workshop; and a series of interviews with eight composers.

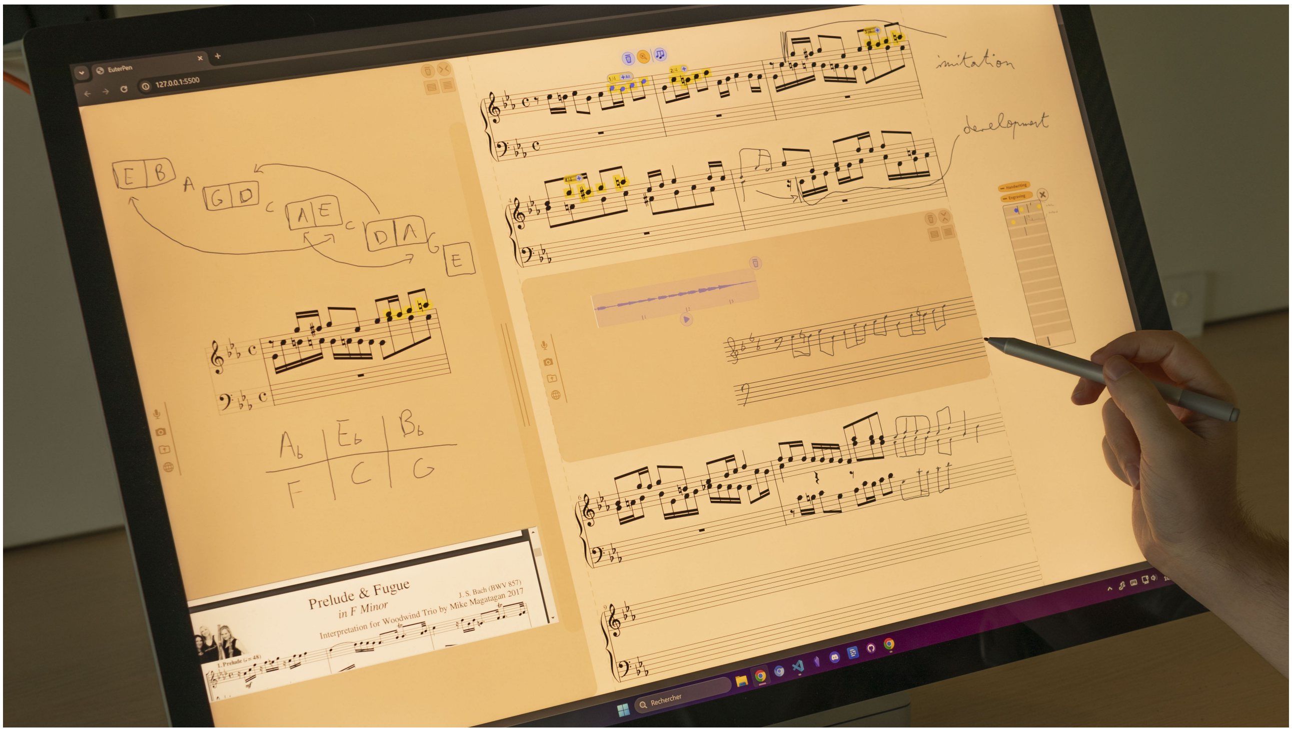


Figure 4.5: EuterPen running on a desktop interactive surface. Areas around staves (called pensieves) are used to freely explore musical ideas while keeping the main score stable, and to store content that supports composers' creative process: documents, audio samples, diagrams. Handwritten and engraved music notations can coexist everywhere, and are both amenable to structured-yet-flexible interactive manipulations, including a tool to search for constrained or relaxed melodic patterns across the whole workspace.

4.4.1 Design Journey

The first design and prototyping phase (4 months) was driven by one high-level goal: to reveal the full potential of interactive surfaces for music score writing.

Once developed, the first EuterPen prototype was used to illustrate a variety of possible interactions to two professional composers who accepted to participate in a half-day face-to-face workshop together with my colleagues and I. We believe it was important to make a high-fidelity prototype (see Section 4.4.2) available to participants even if we were still early in the design process, as composers are not necessarily very familiar with interactive surfaces. We wanted composers to be able to experience the new possibilities offered by this technology first-hand, enabling them to actually manipulate music notation with pen and touch. Discussions during the workshop were structured along different themes. First, the lead author gave an overview of EuterPen. Then, for each theme, a series of features and interaction techniques were showcased to the composers using video recordings, leading to discussions about the relevance of the feature and how it was mapped to interactions. EuterPen was running on a Microsoft Surface Studio 2 in the workshop room, and composers were encouraged to play with the system and try the features whenever they wanted. We made it clear to composers that despite the high fidelity of our prototype we were still at an early design stage, and that they should feel free to question and rethink what they were shown and propose new ideas. In addition to these thematic discussions, a 1-hour brainstorming session was dedicated to the specific topic of *Copy & paste of music notation*. Composers and designers were invited to first produce around 10 ideas each, on separate post-it notes. After 15 minutes the resulting ideas were shared orally or by demonstrating how this would work in front of the EuterPen prototype. The entire workshop was recorded (audio and video) and later transcribed by the lead author. A thematic coding [169] of those transcripts and recordings was performed by Dr. Catherine Letondal and I.

Following this workshop, we entered a second design and prototyping phase (4 months) during which we developed new features and iterated upon existing ones based on the feedback from the two composers and new ideas that arose from the discussion. This led to the second version of EuterPen, which we used to validate our approach through interviews with a broader set of professional composers. Composers, as many creative professionals, are typically very busy, and we chose to conduct interviews online so that they could participate without traveling to our lab. We conducted 8 interviews, all performed using a videoconference tool. An interview lasted between 45 minutes and 2 hours 30 minutes (1 hour 40 minutes on average, more than 13 hours total). Information about the composers' profile is available in Table 4.3. As we did for the workshop, the key features and interactions of EuterPen were recorded and grouped thematically into coherent feature sets. Those were made available to the interviewees on the Web, embedding the videos into a Web page together with textual explanations. That Web page was shared with the interviewees several days before the actual interview took place, so that they could familiarize themselves with the different features. The Web page was actually a form, that they could use to record early comments and ratings about individual features at their convenience. During the interview, this early feedback was revisited as part of the discussion between the interviewer and the composer. I conducted the interviews, and together with Dr. Catherine Letondal we performed the transcription and thematic coding of those recordings. Ratings are reported by thematic group in Figures 4.7, 4.9 and 4.12.

Finally, we entered a third, shorter design and prototyping phase (1 month) to make final improvements to the prototype. These different stages constitute EuterPen's design journey, which we discuss in Sections 4.4.4–4.4.6, emphasizing the main insights from the different stages of the process.

4.4.2 Software Implementation

EuterPen is developed as a Web application and can run in any browser that supports the W3C Pointer Events API [26]. It uses the VexFlow music notation API [36] to render scores as SVG elements [13], that can be imported from MusicXML documents with OpenSheetMusic [175]. The programmatic manipulation of the SVG-rendered music notation elements and all UI components is coded in JavaScript with D3 [21]. Handwritten music notation recognition is handled via a custom version of MyScript [117] configured with a specific grammar. This recognition service runs locally as a Java servlet that EuterPen queries to interpret digital-ink input.

EuterPen has been implemented to experiment with, and demonstrate, novel ways to support composers in their creative process. As such, we did not aim to develop a full-featured score editor but rather a functional prototype that can import music scores and let users modify them or write music from scratch. Section 4.4.3 discusses those novel ways to support composers, focusing on novelty and design choices rather than describing every single feature.

4.4.3 Design Principles

The participatory design workshop and evaluation interviews have confirmed the need for flexibility and have shed light on an additional, complementary principle: stability. On the one hand, composers find it important that the editing decisions they make are not overridden by the music notation program. For instance, inserting bars, adding notes or annotations can have dire consequences on the global organization of the score. But, on the other hand, they find it as important that the program allows them to explore ideas freely without disturbing what is already settled.

How EuterPen follows the guidelines of Section 4.3 and above principles is discussed in the next three subsections, organized according to three themes: breaking the score's linear structure (Section 4.4.4); breaking the score's homogeneity (Section 4.4.5); and breaking down musical elements (Section 4.4.6). In each subsection we also explain where the main ideas originated from and report on the feedback we gathered from composers.

A design scenario (see Scenario 1 below) illustrates how composers can benefit from those guidelines and principles as implemented in EuterPen. This scenario is based on the interviews conducted during the evaluation phase. It is built using composers' feedback about EuterPen interactions – feedback that is rooted in their actual work practices.

Scenario 1 – *A Composer's Journey* – illustrates EuterPen's features using examples of composers' actual practices gathered from the evaluation interviews. A (*) indicates suggestions originating from composers who participated in the interviews, but not yet implemented in the EuterPen prototype.

Chris has started to think of a piece in the fugue form inspired by J.S. Bach, that he has to work on for his composition class. While he is trying some rhythmic patterns on the piano, he decides to record a few audio samples on his smartphone.

A few days later, he decides to create an EuterPen score to explore and shape his piece. He opts for a 4/4 E-♭ signature and writes the first two bars with the pen. To check that it sounds as he imagined, he makes the play gesture with the pen to hear those bars (see Figure 4.7-c and Figure 4.10-c) and is happy with the result. Next Chris wants to insert one the samples recorded on his smartphone a few days ago. He opens the inter-staff *pensieve* (as in Figure 4.7-c) and loads them there (similar to Figure 4.6.c). One of those samples fits very well as a continuation of the first two handwritten bars, and he drags-and-drops it on the staff (e.g. Figure 4.9-b). He can then listen to the whole content, EuterPen seamlessly playing handwritten and engraved notation, as well as audio samples that have been inserted on the staves. Chris then writes a few words above one of the bars to remind him about an idea he had while exploring ideas on the piano, which was about combining background sea sounds with the pattern (he quickly sketches a diagram representing this idea using the pen, as in Figure 4.5). He also loads one of J.S. Bach's fugues as a PDF in the global *pensieve* on the left of the staves (see also Figure 4.5).

Several days later, Chris has completed the structure of his piece in three parts, partly filled with two voices that he has already engraved. Using EuterPen, his teacher annotates the score using the pen (similar to Figure 4.9-c) and makes suggestions such as adding a central part aimed at developing the first theme. She also confirms that Chris could use the 1st voice as a model for the 3rd one.

Back home, Chris selects the 1st voice (as in Figure 4.10-a) over 3 bars and duplicates it before moving it to the 3rd voice. But he has to adjust a few notes to comply with the rules of harmony: using the pen he selects the notes, changes their height, and slightly adjusts their horizontal position so that the score remains legible (similar to Figure 4.12-c). Chris then proceeds to the central development that the teacher suggested. He first explores variations on the patterns using the inter-staff *pensieve*, where he can freely instantiate short staves and play them separately (Figure 4.7-a). He then discovers a better rhythmic form that would make sense for the whole piece: he selects the elements involved in this pattern (similar to Figure 4.12-a) and issues a search over the entire score – not just notes on staves but *pensieves* and annotations as well (*). This highlights all pattern occurrences throughout the workspace. Those highlights are also visible on a minimap that shows a global view of the document (see Figure 4.11). Chris puts all occurrences back in handwritten form and edits them all at once.

Going back to the central development, Chris wants to check Bach's original score that he imported as a PDF, and copy a fragment of it in the *pensieve* between two staves – close to the part of his own score that echoes this passage. Copy-&-pasting this fragment from a foreign source automatically creates a hyperlink back to the original material (*). Such *provenance* information will prove useful to him when he revisits his composition in the future. Finally, Chris inserts 30 bars at once to prepare the development section, using a gesture similar to Figure 4.10-c. He fills those by copy-&-pasting the notes drafted in the inter-staff *pensieve*. This leaves several bars incomplete (as in Figure 4.7-a) but that is fine. Happy with this step, Chris feels like he now needs some time to reflect on this first draft. He closes EuterPen, confident that he will be able to quickly find the places that still need some work in his score as he has left them in handwritten form whereas the finalized bars have been engraved.

4.4.4 Breaking the Score's Linear Structure

A musical piece may unfold temporally, but the creative process that yields it is most often non-linear. This is a source of tension as: on one hand, the notation program should help composers comply with the rules governing

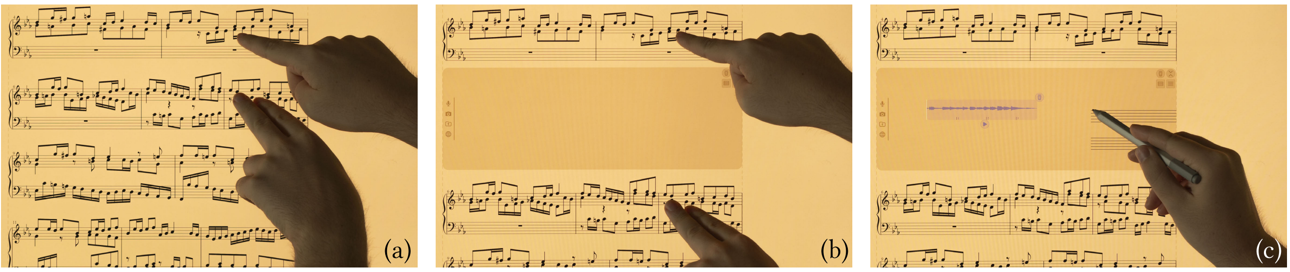



Figure 4.6: Carving a pensieve by (a-b) stretching space between two staves using a simple multitouch gesture, and then (c) populating that space with multiple objects using drag-&-drop. An audio sample has been dropped already, and the user is about to place an empty staff to write music on.

the piece's temporal – inherently linear – structure; but on the other hand, it should let them freely explore and arrange different ideas by supporting a very flexible, non-linear editing workflow.

Existing music notation programs are clearly focused on enforcing temporal-structure well-formedness, providing composers with a workspace organized strictly as a linear sequence of bars forming staves or systems of staves that all abide by the declared time signature. The constraints imposed by those programs do not apply to the primary music notation only, but prevent composers from inserting other pieces of information that are key to the composition process: text annotations or foreign objects such as images that are not meant to stay but support the creative process. Composers have to cope with these constraints, when an exploratory process rather calls for those constraints to be lifted, if only temporarily. When possible, composers will cheat the program: for instance, adding dummy bars at the end of a score to test ideas that are not meant to be part of the final piece. However, many constraints cannot be circumvented, leading composers to resort to other means to capture their ideas, such as pen and paper. This has a cost though, as they will then have to juggle completely disconnected media and workspaces.

We address this global concern by designing EuterPen not as a regular *score editor*, but rather as a *music processor*.[3] While EuterPen has composers write music notation on staves as any regular score editor does, composers can interactively carve spaces between staves, that can hold a variety of contents. Two spaces coexist:

- the regular **temporal composition space** with a linear flow of staves, as found in all music notation programs;

[3]A termed coined by LA during their interview, drawing an analogy with *word processors* that are more flexible than structured document editors and enable authors to mix different types of contents.

- and canvases, named **pensieves**,[4] that composers can instantiate and populate freely with a variety of contents.

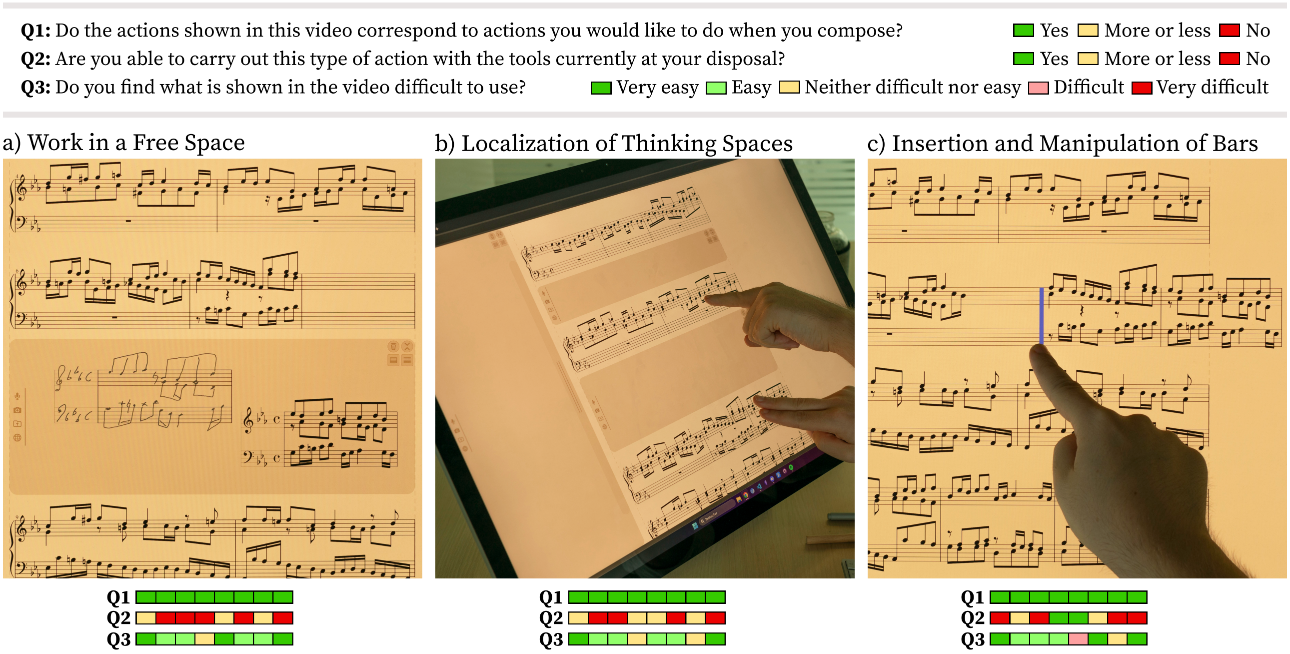
Pensieves can be used to store and retrieve any material relevant to the creative process (handwritten text annotations, drawings, pictures, audio samples and video clips). They can also be used to experiment with musical ideas, instantiating blank staves to write or copy music notation on – see Section 4.4.5. As such, pensieves exist in-between and around staves, and can be opened and closed at will.

Carving a pensieve between two staves in the temporal composition space involves holding one staff with a finger, and pulling the other staff with two fingers. This type of pensieve remains tightly integrated with the temporal composition space and scrolls along with it. A global pensieve can be shown to the side of the score as well by dragging it like a drawer. That global pensieve spans the full height of the EuterPen workspace and can be scrolled independently from the staves of the temporal composition space.

Pensieves let composers organize content *around* staves. But EuterPen also aims to give them more flexibility *inside* the staves themselves. The systematic enforcement of temporal constraints in bars represents a major source of frustration and impedes the creative process [32]. EuterPen addresses this key issue by relaxing these constraints. Staves can hold a mix of structured (engraved) spans and unstructured ones consisting of arbitrarily-long sequences of notes and rests. When writing music, composers can – but are not *required to* – draw bar lines, and then engrave or delete them at will. EuterPen also makes insertion more flexible in order to better support the creative process, which is highly non-linear, as mentioned before. A caret gesture [116] performed with the pen (see Figure 4.10-c) will push existing notes to make space that can then be used to write new notes. As the layout of bars on staves is something that composers consider important, the insertion strategy can be parameterized on-the-fly: adjust the amount of space depending on the length of the caret gesture; after lifting the pen, choose to add space before (push content upstream) or after (push content downstream) the insertion point; and most importantly choose whether to automatically generate bar lines (which helps remain within a given duration) or not (which leaves the duration unconstrained).

EuterPen lets composers make further layout adjustments to engraved content, leveraging the structural information obtained in the process: spreading or packing notes with a two-finger pinch gesture, adding space to a bar by dragging the bar line with the pen. EuterPen does not attempt to optimize layout automatically as other score editors do, but rather gives

[4]Following the suggestion of a workshop participant, who drew an analogy with the thought spaces described by character Dumbledore in the *Harry Potter* books series.



Answers to Q2 color-coded red or yellow are to be interpreted in favor of EuterPen, as they indicate operations impossible or impractical in the score editors that interviewees currently use.

Figure 4.7: Breaking the score's linear structure: participants' evaluation of the features demonstrated in the videos shown during the interviews.

composers the possibility to resize bars according to music notation rules on demand.

Beyond staves and pensieves, which constitute two distinct spaces with different purposes, EuterPen also lets composers make annotations anywhere in the workspace – in all areas, both temporal composition space and pensieves. Composers create those annotations seamlessly without the need to switch modes, as explained in Section 4.4.5.

Feedback

As illustrated in Figure 4.7, interviewees were very enthusiastic about EuterPen's way of breaking the score's linearity: canvases for free-form work and the flexibility in handling bars were unanimously judged to be useful (Q1). Interviewees did not think these exist in their current music notation programs, except for the insertion and manipulation of bars, which two composers felt they can already perform as they wish (Q2). Composers found the demonstrated interactions easy to use with the exception of one composer who found the caret gesture to insert measures potentially difficult to perform (Q3).

The idea of providing a blank canvas for composers to freely arrange and edit measures outside the score stemmed from the opportunities identified by Cavez *et al.* [32] and eventually grew into pensieves during the workshop. As C_2 put it: *"it would be great to be able to copy, how shall I put it, copy and paste, but outside our score"*, C_1 then suggesting this could be a *"reserve of ideas, pensieve or clipboard"*. The first EuterPen prototype that C_1 and C_2 saw during the workshop only allowed carving pensieves between staves. The concept was thus expanded to include the global pensieve (on the left in Figure 4.5), which is not tied to a particular place in the score but rather remains fixed on screen, serving as a global storage and annotation space.

The relevance of pensieves was further confirmed during interviews: *"And yes, it leaves a little bit of... a little bit of a laboratory in the middle of the score"* (BF). It is not just about *"having everything already on screen rather than travelling around in windows,"* but also about having space, *"a bit like having cards and moving them around"* (LA) instead of having to create extra bars, *"which is hell, because then you have to take them away"* (AO). Composers found multiple uses for pensieves. They can be simple temporary work areas, for example to *"switch the left and right hands"* (LA). Composers often saw them as a space to note comments, draw, but also to explore musical ideas, for example to *"try out fifteen different rhythms"* (LA). Staves created in pensieves (e.g. Figure 4.5 or Figure 4.6-c) are fully functional: they can be interpreted, played back, and *"re-injected into the score at a later time"* (DP), a feature that LA was enthusiastic about: *"It's really incredible [...] adding staves on the fly underneath. Wow. Fantastic [...] You draw five lines by hand to make the poor man's staff, and now the computer does it itself. It's much more practical."*

Pensieves could even be used to build a structured library of ideas: *"I can imagine this as a sort of idea box where you can have places where you have a zone with rhythmic ideas, a zone with ideas for a melodic sequence and you can take one or the other"* (C2). This notion of a library extends beyond music symbols on staves to other creative resources: musical references or sources of inspiration (as in Figure 4.5), as C1 described during the workshop: *"You can get the whole idea bit by bit. You don't have time to go into all the details of the thing, but then you have to note down as many resources as possible. If, for example, I want to take inspiration from Steve Reich, then I'll put Reich's score on the page so I don't have to go looking for it."* It can also be a teaching tool, as AO points out: *"There were times when my teacher said to me: 'Yes, but that, you have to look at Debussy, look at* La Mer *by Debussy. Look at all the string models'"*. Or as OS says: *"to make exercise sheets perhaps."* He sees pensieves as a tool to *"import whatever you want. Images, text. Links to pages like that, [...] stories about music or anything else that might relate to the study of the orchestra, for example instruments in the orchestra."*

The greater flexibility obtained from the relaxation of structural constraints within measures was also received quite positively. Composers can experiment freely with melody and rhythm without having to worry about the notation's formal correctness, which fosters creativity: *"There's no more alteration, there's no more rhythmic signature in fact? You can do what you like, you're free, aren't you? That's great. The example you did on the side, can you add beats, can you do a nine-beat bar like that, even without numbering it?"* (YT), echoed by (BP) who observed that *"the bar we're going to insert can be of any rhythmic signature"* suggesting he would rather *"define afterwards whether it's identical to the preceding bar or whether it's a completely different rhythmic signature."* BF further added: *"There may be an irregularity at some point in the beat, a change of time signature, for example, or a change at the end of a phrase that runs counter to the beat"*, which may lead to *"the possibility of adding an extra beat to a bar"*. Similarly, composers commented positively on the management of empty space in bars. While other programs automatically insert rests to keep a bar well-formed, EuterPen leaves composers free to write down their ideas unimpeded: *"So typically, when you've deleted the two beats, it hasn't put anything in their place. I like that, that there's nothing there"* (LA).

Composers also appreciated not having to worry about bar layout at the start of the writing process. As AO puts it: *"when I'm composing, I'm spending time doing something and I, I shouldn't be spending time doing [layout]. In fact, ideally, I should be composing."* echoing observations from Bennett [14]. Layout stability – avoiding automatic optimizations – was also considered important: *"Well yes, that's very practical because everything stays there. And then you do the layout again, but after you've finished the whole passage and not while you're adding notes one after the other"* (LA).

4.4.5 Breaking the Score's Homogeneity

To better match composers' mental model and creative process, EuterPen supports two representations of the primary notation that afford different manipulations. The first representation uses regular, beautified symbols as found in other music notation programs. The second notation is the composer's own handwriting. The two can coexist seamlessly on a stave and even be intermingled in the same bar. The former representation, because of its engraved look & feel, suggests fully-developed, finalized material; while the latter rather suggests material that is still work-in-progress and more likely to change. This is further reinforced by the differences in how composers manipulate the two representations. Handwritten symbols can easily be deleted by flipping the pen and using the eraser – as on paper. On the contrary, engraved symbols must first be selected before they can be deleted. Composers can thus easily jot ideas down by drawing symbols on the score, and they can as easily discard them with the pen's eraser, without running the risk of inad-

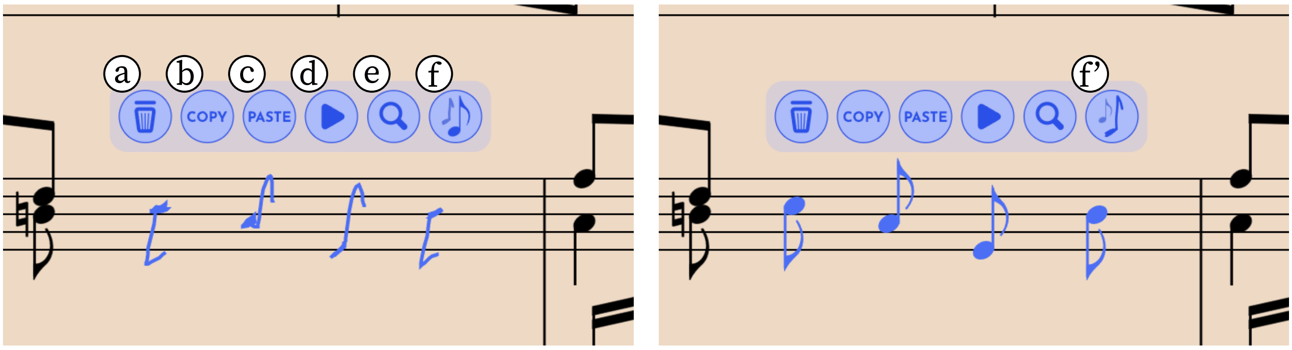


Figure 4.8: A note selection and the associated contextual menu: a) erase selection, b-c) copy-&-paste selection, d) play back selection, e) find occurrences of the selection, f/f') switch between engraved and handwritten representations.

vertently modifying or deleting engraved symbols nearby.

The two representations – engraved and handwritten – can be used on all staves, in both the main composition space and pensieves. One important design decision we made was to leave handwritten music symbols seemingly uninterpreted by default. Handwritten symbols necessarily have to be interpreted when composers ask to engrave them (Figure 4.8-f), but this interpretation actually takes place in the background even before that, as soon as they get selected (with a lasso selection gesture). This lets composers perform operations that would typically only be possible on engraved material on handwritten notation as well – while preserving their *work-in-progress* representation. Such operations include : copy-and-pasting the selected notes (Figure 4.8-b & c); listening to them (Figure 4.8-d);[5] and searching for other occurrences of the same pattern throughout the score (Figure 4.8-e and Figure 4.9-a) – both engraved and handwritten. Finally, the conversion from handwritten to engraved notation is not a one-way transformation. Composers can actually toggle between the two representations at will (provided the selected symbols were actually input with the pen in the first place), which can be useful for instance to visually revert a passage to a more *work-in-progress* look & feel.

The decision not to interpret handwritten symbols immediately was taken for two main reasons. The first reason was to give composers as much freedom as possible, preventing the parsing process from forcing a particular way of writing on composers and thus avoiding the pitfalls of greedy input interpretation commonly found in other music notation programs [32]. The second reason was to enable three very different types of ink-based input to coexist without resorting to mode switches: handwritten music symbols; arbitrary secondary notations such as text and drawings (Figure 4.9-c); and a variety

[5] Such auditory control is key to support elaborate tasks such as transcription, incrementation or modification

of gesture-based commands (see Section 4.4.6). Composers can seamlessly perform any of these thanks to the following disambiguation strategy:

- any pen mark that 1) has been initiated in empty space, and 2) does not match one of the three predefined gestures (Figure 4.10), will remain uninterpreted and treated as a simple annotation (secondary notation);
- any pen mark that has been initiated on a staff will be considered a candidate music symbol for interpretation, if selected later;
- if the trace currently being inked matches one of the predefined gestures, the ink color will change, providing feedforward to composers about how the gesture will be interpreted if they lift the pen then (for instance, the pin-to-play gesture in Figure 4.10-b);
- if this is not the interpretation they intended, composers can continue inking and lift the pen later when the gesture no longer matches any predefined one, which will be indicated by the ink reverting to its default color.

.

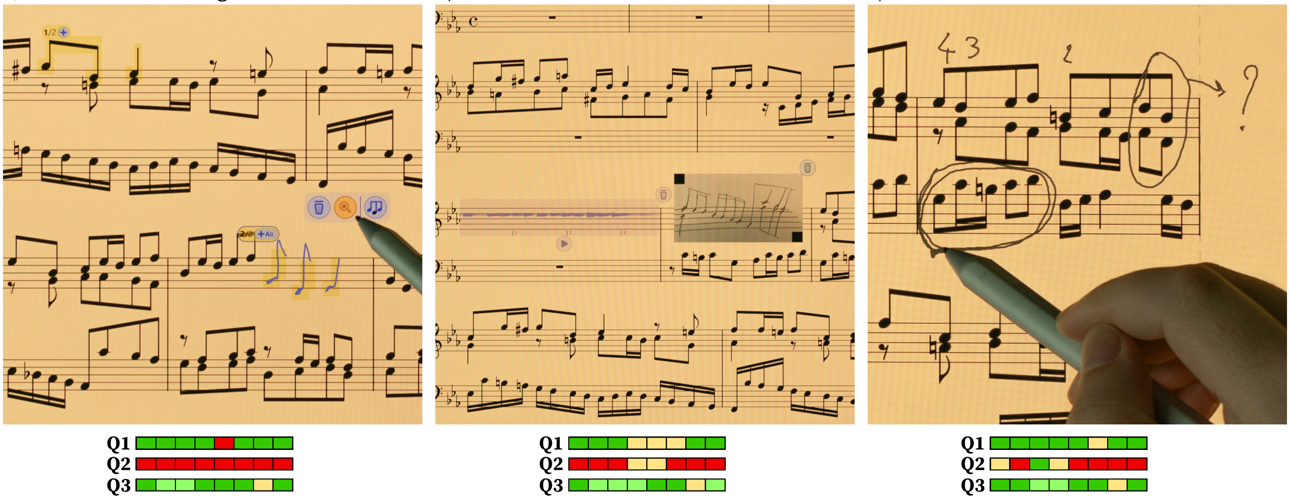
This strategy works all over the EuterPen workspace and lets composers add secondary notation not only in pensieves but anywhere on staves as well, more tightly integrated with the primary notation – for instance ink marks made to circle and link different passages on the score. Such secondary notation is actually not limited to handwritten text and drawings, but can be much more heterogeneous. Composers can drop different media types directly on the staves (replacing some bars) or very close to the staves. The list of supported media types is informed by results from Cavez *et al.*'s study [32] and by our workshop: an audio recording made on-the-spot using the tablet's microphone; a picture taken on-the-spot using its camera; a document (PDF, image, audio, video) imported from the file system or linked from a URL. The purpose and relevance of these secondary notations will evolve as the composition progresses. Composers can thus easily move, resize, show and hide or entirely discard them at will.

Feedback

As illustrated in Figure 4.9, interviewees were very positive about EuterPen's way of breaking the score's homogeneity: mix handwritten and engraved notations, insert other media such as audio samples on staves, put secondary notations anywhere, were possibilities deemed useful by a majority of composers (Q1). Those features are largely absent from current music notation programs (Q2), and seemed generally easy-to-use in EuterPen (Q3).

Q1: Do the actions shown in this video correspond to actions you would like to do when you compose? — Yes / More or less / No
Q2: Are you able to carry out this type of action with the tools currently at your disposal? — Yes / More or less / No
Q3: Do you find what is shown in the video difficult to use? — Very easy / Easy / Neither difficult nor easy / Difficult / Very difficult

a) Music Handwriting and Conversion b) Cohabitation of Different Notations c) Score Annotation



Answers to Q2 color-coded red or yellow are to be interpreted in favor of EuterPen, as they indicate operations impossible or impractical in the score editors that interviewees currently use.

Figure 4.9: Breaking the score's homogeneity: participants' evaluation of the features demonstrated in the videos shown during the interviews.

Secondary notations are useful not only to make text annotations but to make freeform drawings as well – what LA calls *"supervised cohabitation"* – that can capture intent, help understand something, give information about musical structure (see Figure 4.5): *"Maderna's* Serenade pour un satellite *is a circular score"*. . . *"musically [. . .] it's illegible"* but *"in a circle you can make choices, you can go left and right . . . [so] the idea of engraving it like that rather than explaining it logically, was really so that it would inspire musicians"* (LA). Drawing could also be a way to *"invent symbols that you're going to put in the score"* (AN) or to support musical ensembles as a way to *"build the score by inserting an idea of physical movement, even explaining it with a diagram, showing everyone's place, with a movement that's taken as you would on a sketch"* (YT).

The idea of integrating music passages in forms other than that of the primary notation (an audio sample, a picture of a roughly sketched passage to be transcribed - see Figure 4.9-b) had emerged during the workshop. Its relevance was further confirmed during the interviews: *"It would be fantastic if... [the system could play] what was engraved, written, and given that there's the MP3 afterwards, given that there's the audio afterwards, we'd really like it to follow on"* (LA). Similarly, C_1 wished they could work with audio recordings of themselves searching for ideas on the instrument and integrate those easily in the score: *"I have a lot of memos in my phone that should be developed, but* *they are in my phone, and I cannot put them in the score."* Several interviewees saw potential in the combination of pensieves and secondary notations as a means to notate and coordinate group work: *"It can be really useful because it transforms the score into a performance"* (DP). On the same topic YT said: *"We're in the process of doing something, creating a show with the double bass orchestra. We're making it by hand. In other words, there are lots of things. It starts with an idea. And then the work in progress happens in rehearsal, with five people, and everyone puts in their two cents."* And since *"the sixth person wasn't there because she's away on business"*, so it's like *"an instruction manual"* for her.

The possibility to make the handwritten and engraved representations coexist had much success.[6] Composers appreciated that many operations could be performed on handwritten passages as well, such as play them, copy-and-paste them, or look for other occurrences (Figure 4.8): *"Searching for the motif to see if you've already written it before beautifying it, I'd use that every day"* (LA). During the workshop, both C_1 and C_2 commented very positively on the possibility to immediately listen to handwritten passages using a simple gesture (Figure 4.10-b) without having to engrave them. But at a more fundamental level, the possibility to switch back-and-forth between engraved and handwritten representations was seen as a means for composers to keep track of where they are in their creative process: *"You don't have to ask yourself the question 'Is this a final version, is this my draft?' "* (LA), because this enables *"visualiz[-ing] very, very easily a passage that needs to be reworked"* (DP). As C_1 observed, it could also help composers keep track of the decisions made: *"Going back in time can really help. Often you get the clean version and can't remember how you got there."* But during the workshop composers also wondered how this would work for engraved notation not originally written by them (for instance, bars imported from a MusicXML file). C_1 emphasized that it would be *"seriously awkward to see the handwriting of someone else"*. In such cases, one option discussed with C_1 would be to use machine learning to train EuterPen on the composer's own handwriting and to have it generate the handwritten representation, possibly involving *ink beautification* and *handwriting generation* mechanisms [160].

While some score editors do enable the annotation of scores, composers saw the possibility to draw and add pictures or audio recordings directly on the staves as a means to support creativity, that could be useful for teaching as well: *"We can even, while keeping what [the student] has written, add something else in manuscript [. . .] something that can go in another direction"* (YT).

[6]One composer found this of little use (Figure 4.9-a), primarily because they found it difficult to draw notes on a screen.

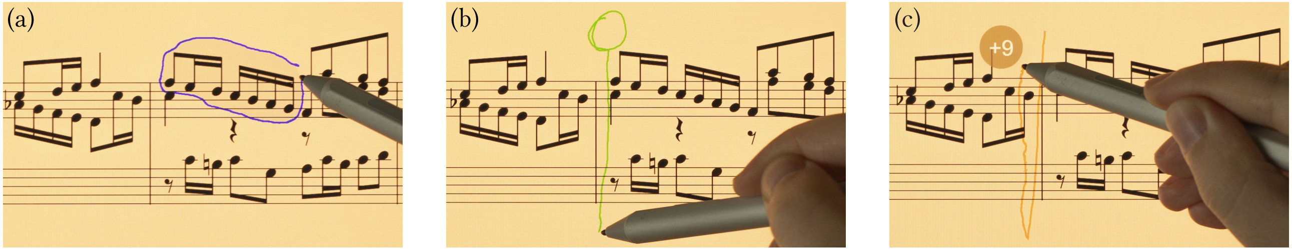


Figure 4.10: The three predefined pen gestures available to composers, that coexist with all direct manipulations on the engraved notation as well as the input of handwritten notation – all without any mode-switch : (a) lasso-to-select; (b) pin-to-play; (c) caret-to-insert (the amount of space to insert being controlled by the length of the upward trace, here 9 bars).

4.4.6 Breaking Down Musical Elements

As discussed in subsection 4.4.5, one of EuterPen's core design principles is to make multiple types of ink-based input coexist without resorting to mode switches: handwritten music symbols, secondary notation, and predefined gesture-based commands. In addition to these, and consistent with current editors, EuterPen should support pen- and touch-based direct manipulation of music notation elements. This introduces further ambiguities that need to be resolved.

Predefined gesture commands. Three actions are triggered by pen-based gestures that are easy to learn and to perform (Figure 4.10). These gestures are recognized by a simple, robust classification algorithm that accounts for variability in gesturing. To be scale-independent, it uses the gap between two staff lines as a reference distance to classify input traces: *lasso-to-select* encloses an area at least $2 \cdot gap$ wide and high, ending within $2 \cdot gap$ of the start point; *pin-to-play* combines a lasso with a vertical trace at least $4 \cdot gap$ high; and finally *caret-to-insert* is a downward vertical trace of at least $4 \cdot gap$ high followed by an upward trace.

Direct Manipulation. Music notation elements that lie on staves can be moved horizontally or vertically to change the pitch or adjust the layout. For instance, when selecting a note head, a small cross will appear (see Figure 4.12-c), suggesting the possibility of direct manipulation in these directions. Dragging along another direction will initiate one of different lasso selections, as detailed next.

The digital pen is an excellent tool to write music symbols. But it is also a very precise selection tool, that can be used to delineate free-form areas. As such it is well-suited to the selection of music elements, which are composed of multiple tiny glyphs densely packed together. The modern staff notation is a multi-dimensional grammar that encodes different auditory attributes of a note (pitch, duration, *etc.*) by combining those glyphs and positioning them precisely on staves and ledges. The most frequent glyphs include the head, stem, flag, beam, dots and accidentals. Some glyphs define a note's auditory attribute, while other glyphs are rather modifiers of that attribute. But regular score editors typically consider notes as entities that cannot be further decomposed. Selecting any constituent glyph of a note will select the whole note, that can thus only be moved (adjusting pitch), copy-&-pasted, or deleted.

EuterPen aims to break this monolithic view on notes and let composers select individual auditory attributes of a note if they want to. This is made possible by introducing novel types of selections. Composers can invoke those novel selection tools seamlessly, still without switching modes. We achieved this by adopting a strategy that disambiguates what to select based on *where* composers start their selection. A lasso enclosing multiple notes will select:

- the whole notation if initiated outside of any glyph – see Figure 4.10-a;
- handwritten notation only if initiated on handwritten symbols, ignoring engraved notation;
- a series of durations if initiated on a rhythm-related glyph (beam, stem, flag or rest) – see Figure 4.12-a;
- a series of pitches if initiated on a note head, dragging diagonally;[7]
- accidentals if initiated on an accidental;
- text if initiated on textual elements such as, *e.g.*, dynamics.

To help composers perform those different selections, EuterPen provides feedforward, dimming the glyphs that are not related to one of the selected auditory attributes until the lasso selection is completed. Figure 4.12-a illustrates this feedforward for a selection initiated on a rhythm-related glyph (in this case a quaver), temporarily dimming note heads and accidentals.

While selecting individual auditory attributes makes little sense for coarse manipulations such as deletion, it provides composers with a whole new set of capabilities for more elaborate actions such as Find or Copy-&-Paste.

Find. Repetitions and variations are an essential part of many compositions [32]. Highlighting the occurrences of a melodic pattern is thus an important feature of music notation programs. EuterPen highlights such occurrences directly in the score, and also shows them on a minimap of the score (Figure 4.11-b), facilitating navigation over the result-set. Composers can then effortlessly add these occurrences to the active selection and manipulate them together – for instance to adjust their relative pitch. Combining this feature with EuterPen's novel selection capabilities actually enables composers

[7]As mentioned earlier, horizontal and vertical movements on note heads are reserved for direct manipulations.

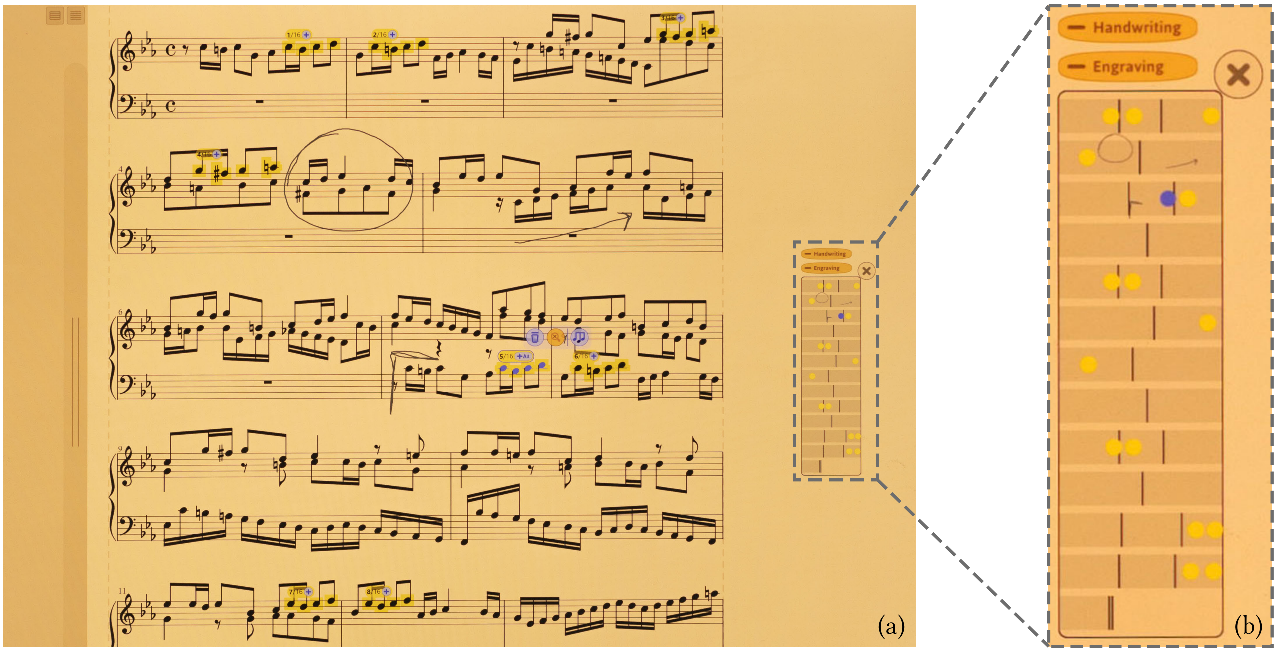
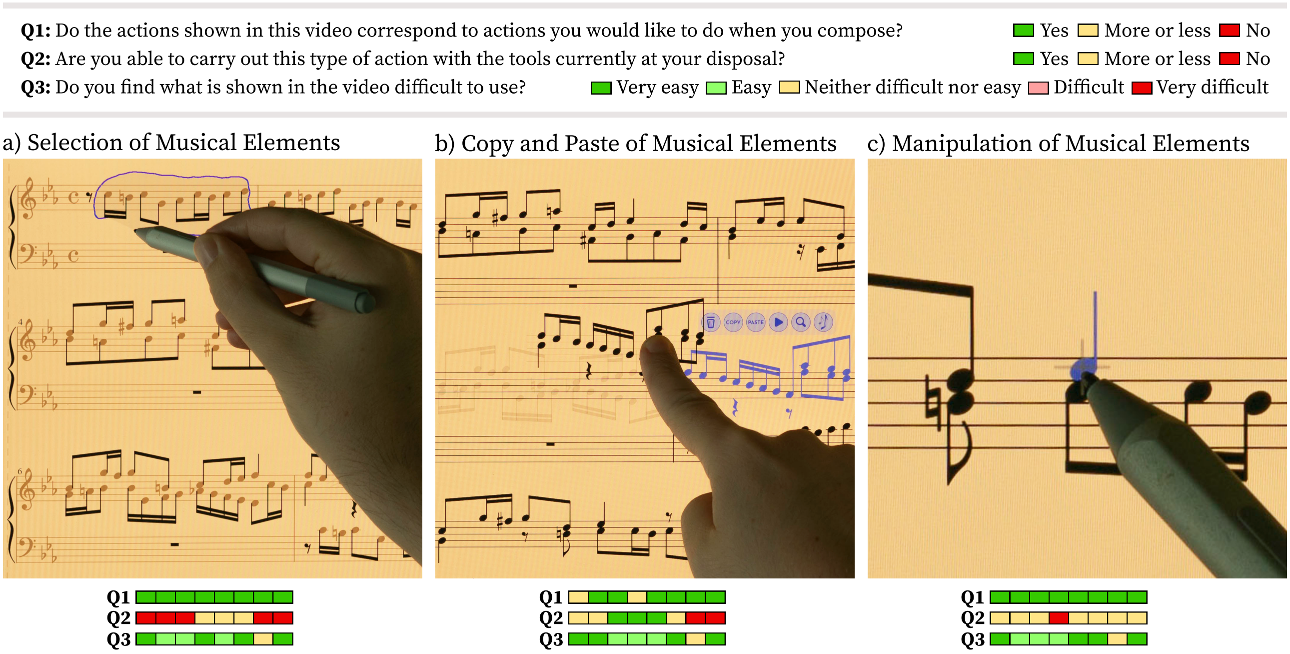


Figure 4.11: Searching for all occurrences of a melodic pattern based on pitch only, relaxing constraints on rhythm. (a) Four note heads have been selected (colored blue, 7th staff), and all occurrences of the same pitch sequence (C-B-C-D) are highlighted yellow, regardless of their duration. Occurrences of the same sequence but transposed are also highlighted (*e.g.*, E♭-D-E♭-F, 11th staff). (b) The sixteen occurrences are also highlighted on the interactive minimap that gives an overview of the whole score. Annotations made on the score are also visible on the minimap, providing composers with navigation landmarks.

to make queries involving specific auditory attributes only. They are able to search for a rhythmic pattern regardless of pitch or, as illustrated in Figure 4.11-a, search for a pitch sequence regardless of rhythm.

Copy-&-Paste. Music has always been about repetition and variation (as AO says: *"Mozart would have been so happy to be able to copy and paste notes! I think he must have copied quite a few..."*), and composers copy-&-paste music notation very frequently. Available in many score editors (Figure 4.12-b, Q2), the possibility to duplicate music notation this way represents a major advantage of computer-based editing over paper. But the way regular score editors implement copy-&-paste is very monolithic and provides little feedforward. It makes it difficult for composers to anticipate what will happen, and sometimes yields unexpected or unwanted results. EuterPen takes a different approach to copy-&-paste. Dragging a pen-made selection with a finger will duplicate it (*drag-to-duplicate*). While dragging, notation currently at the destination will fade-out almost completely to show composers where the duplicate notation will be placed if they drop at this moment, as illustrated in Figure 4.12-b. Another way to control precisely where to paste is to ex-



Answers to Q2 color-coded red or yellow are to be interpreted in favor of EuterPen, as they indicate operations impossible or impractical in the score editors that interviewees currently use.

Figure 4.12: Breaking down musical elements: participants' evaluation of the features demonstrated in the videos shown during the interviews.

plicitly select what to replace (*select-to-replace*). Once a selection has been copied, composers can select what symbols they want to replace with what is in the system's clipboard.[8] A tap on the PASTE button in the selection contextual menu (Figure 4.8-c) will then effectively replace the notation. Again, combining this feature with EuterPen's novel selection capabilities opens new possibilities. By restricting the selection to one auditory attribute only (pitch, duration), composers can for instance copy a sequence of pitches and paste them on another rhythm than the original one; or conversely they can copy a rhythm and paste it on another pitch sequence than the original one – all without having to erase and rewrite the entire sequence.

Feedback

As illustrated in Figure 4.12, interviewees were very enthusiastic with EuterPen's innovative way of breaking down musical elements to select, copy-&-paste and manipulate them with precision. Composers were almost unanimous in recognising these features as useful (Q1), although half of them have

[8] Copied content is not necessarily meant to replace existing content. Drawing a small lasso in an empty spot will *insert* – rather than *replace* – the notation at this location.

already found workarounds with their current music notation programs (Q2). They found the way EuterPen supports these features easy to use (Q3).

Selection plays a key role in many operations and is of major concern to composers. Interviewees found EuterPen's extended set of selection tools useful (Figure 4.12-a) and shared their thoughts about how they would use them in their work. For instance, OS would like to select melodic voices with ease *"to erase and replace with something else because I made a mistake. It's often the intermediate voices in particular that I have problems with. It's true that you often have to erase everything and start all over again."* YT saw potential in auditory attribute selections, for instance to try different rhythms for a given pitch sequence: *"If we have a melody that we like, we can change the rhythm. I've sometimes started with a sequence of notes, and thought to myself 'what can I do with this? Here, I'd like it to have this groove.'"* But also the other way around, to manipulate only the melody: *"We'll change the pitches and keep a form that's close by changing the notes [...] I'll take this melody. So that will be my wink and reference to that melody, but I'm going to reverse it."*

Combining search with auditory attribute selections to highlight full but also partial matches for a given melodic pattern was also commented positively about. Modern classical composers seek structural clarity and our interviewees talked about how they could use relaxed queries constraining rhythm only: *"When you're doing an orchestration with lots and lots of content, it can be nice to say 'Wait, let's see all the people who do this rhythm'"* (AO). Similarly, for melodic patterns: *"[...] you search for the subject at the beginning and see it pop up in the score, and that's it. Your analysis is done. Even hidden motifs will appear, things that can be overlooked because they're placed between voices two and three, they'll appear on their own"* (LA). DP and BF, both jazz composers, give even more importance to *"series of notes"* and *"series of rhythms"*: *"I only work with small pieces of material that I repeat and transform, and which end up in different forms, sometimes very hidden in different parts of the score"* (DP). BF explains how he would use the search feature to find a formula to edit: *"I know that I've put this formula in the middle of a development and I want to find it again. Maybe it's quicker that way, since it's highlighted. So it provides a focused reading of the big picture."* Composers including AN, LA and BF also expressed the need to extend the search's scope: to be able to search in pensieves and to include secondary notation as well. The minimap (Figure 4.11) was first suggested during the workshop, and requested again by several interviewees, but only added to EuterPen in the last prototyping phase. It shows not only melodic pattern matches during a search, but bar lines and handwritten annotations as well, as these provide strong navigation cues.

Feedback about direct manipulation of the primary notation was positive (Figure 4.12-c). BF emphasized the need for fine-grain manipulation to improve legibility: *"when the voices are very, very close together, simply to avoid* *blots. And if you have accidentals in addition to this crowded writing... it gives you clarity."* While most composers can already adjust symbol placement with their current editor, some complained about the rigidity of such manipulations and their lack of stability: *"It moves all the notes. For example, I often have a problem with arpeggio notes that, you know, immediately change the dimensions of the score. Just by having this arpeggio, you end up with a bar that's the size of two bars. And then, typically, every time I want to move something, it's going to move the whole score"* (AO). Interviewees also shared ideas about how to further improve direct manipulation of the primary notation, as briefly discussed in Section 4.5.

During the workshop's brainstorming session, C_1 and C_2 both imagined that sequences of notes could be dragged outside of the score, and dwelled on the concept of a rich-yet-straightforward manipulation of specific auditory attributes, whether at the time of copy or paste. This inspired our alternative proposals to classic copy-&-paste, which were well received by interviewees (Figure 4.12-b). Commenting about *drag-to-duplicate*, AO observed: *"I like the fact that you can already perform an action without having to click on a button,"* BF adding: *"This movement, we did not have this ease with the editors. It had to be copied... Moving is complex because you often have to cancel a measure to put something back in [...]. [In the meantime you need to put] rests, because an empty measure is not tolerated by the software."* Interviewees described different uses of copy-&-paste combined with auditory-attribute selections. Leveraging pensieves, AN saw potential for *"a sort of idea box where we have a zone with rhythmic ideas, a zone with melodic progression ideas and we can take one or the other."* LA, who insists on the importance of stability of the score, further stresses the importance of spaces where he can work on duplicated elements before integrating them into the main temporal flow: *"It's a good thing that the pseudo-final material, or the engraved version in any case, is relatively fixed, so that we don't tamper with it too much, but rather copy [some of] it into draft spaces and work there. That way, there's a version that moves less, otherwise, in fact, we'd be, I think, very tempted to throw everything left and right. [Things] would be moving all the time."* Other composers (BF, CI and OS) also expressed interest for auditory attribute copy-&-paste, for instance to change the rhythm of a melodic sequence, acknowledging how easy EuterPen makes such operations: *"This is something we do regularly when there are repetitions of formulas [...] except that we often take only the rhythm. We take the formula and then edit the notes one by one, on the formula we've just copied. So it's a lot more laborious"* (BF).

4.5 Discussion

Talking about Euterpen, one of our interviewees said: *"It's not just sand. [...] You can add a lot of different things in a lot of different places, but it's all still* *linked to something that's constantly there and that you can reframe, in cases where you've expanded it a bit, which is the score that you're working on. And so, around this thing that remains, you can really develop your thinking and organise your work. A supervised cohabitation"* (LA). This comment captures the balance that EuterPen seeks to strike between flexibility, structure and stability, as discussed in the previous sections.

Flexibility can be opposed to structure. It can also be opposed to stability. But in our context structure and stability are two orthogonal concerns. Regular music notation programs are heavily structured, imposing many constraints on the notation and its manipulation. But this structure does not contribute to the score's stability, and can even play against it, structured editors sometimes moving notation automatically to enforce syntactic or layout rules.

The tension between flexibility and structure primarily exists within staves, when editing the primary notation. Structure is what enables advanced selections, copy-&-paste and direct manipulations on (groups of) notes, melodic pattern matching, well-formedness checks. But structure is also what separates writing music in a score editor from writing music on paper. By systematically enforcing the notation's rules, regular music notation programs unnecessarily restrict composers' means to capture their ideas, when those creative *professionals* most often know what they are doing and would prefer to fix syntactic and structural issues later. EuterPen aims to balance structure and flexibility, temporarily relaxing writing and editing constraints to foster creativity. This is for instance achieved by letting users decide when to interpret inked marks to perform advanced operations [148, 179]; or letting them turn their notation back into an "work-in-progress" state at will [195], instead of forcing a more permanent choice at the time of creation. Another way of relaxing constraints while keeping structure within reach is to trust composers with more control over the layout of symbols and measures. EuterPen allowing composers to take the time and space they need to work [150, 198], providing them with the means to trigger automatic layout optimization as they see fit.

The tension between flexibility and stability takes place at a more global level. While this concern was part of the initial design ideas, the brainstorming session on copy-&-paste during the workshop emphasized its importance, which was then confirmed during the interviews. Composers want to be able to operate in the way that suits them best at a given stage, without having to change modes, without having to break what they have already built, with free spaces at hand in which to experiment. To paint a metaphor, composers we worked with ask for a tool that will let them play in sandboxes while preserving the castles they have already built. In that sense, pensieves serve two purposes: a means to store material that supports their creative process; and a means to freely explore creative ideas while preserving the stability of their work. As such, we approached the design of pensieves as canvases that can be instantiated on-demand in different places, integrated in and around the structure of the score [8] – as opposed to having a single monolithic canvas that better suits other types of tasks such as sketching [88] or expressive visualization authoring [194, 146].

One important consideration about pensieves is that participating composers did not see these flexible spaces as unstructured, but quite the contrary. Pensieves let composers drop and freely organize heterogeneous pieces of content, but they offer the same interpretative functionalities as the temporal composition space. Some composers actually went further, suggesting that pensieves could feature "drawers" to organize content, or that they could be used to organize performances of their piece. Pensieves should really be understood as spaces complementing the temporal composition space, not competing with it.

4.6 Conclusion

RQ2: How can pen and structure be articulated in order to promote both efficiency and flexibility on interactive surfaces?

The research question driving our investigation was RQ2 applied to the case of music notation programs: how can pen and structure be articulated in order to enable both efficiency and flexibility with music notation programs on interactive surfaces?

We started this investigation by interviewing nine professional music composers to understand their creative process and their use of music notation software. We gathered high-level observations through thematic coding to capture their thought process and creative intentions, and we analyzed their work process in detail, relying on the "Cognitive Dimensions of Notations" framework adapted to music notation programs (see Section 4.2). These two complementary analyses confirmed that music composers face usability challenges when using these programs, due to their contradicting needs for efficiency provided by structure and flexibility inherent to the creative generation of ideas. We thus elicited three guidelines exploring the potential for interactive surfaces to address this tension and better support composers' creative process: (1) breaking down musical elements, (2) breaking the score's homogeneity, (3) breaking the score's linearity. These are three ways for the user to **temporarily break the structure** of the score for more flexibility, while keeping the structure within reach for efficient manipulations (see Section 4.3). Building on these guidelines, we engaged in the first prototyping phase of EuterPen, a music notation program leveraging the pen opportunities for precision and expressivity. This first prototype was explored by two professional composers during a face-to-face workshop, and then refined based on their feedback. The second iteration of EuterPen was then evaluated by eight professional composers during individual interviews, leading to a positive reception of the tool and its features. Composers were eager to break away from the rigidity of regular music notation programs while still being able to perform advanced editing with the pen. A third and final iteration on the prototype led to the version of EuterPen presented in Section 4.4. This process also emphasized the importance of *stability* over the content of the score when breaking the structure, which represents an additional design consideration for developers and researchers.

4.7 Future Work

We designed EuterPen according to an iterative process with composers, including an online questionnaire in the final stage where nine composers were invited to provide feedback on EuterPen's features. Although composers reacted very positively in this first evaluation, future work should observe how composers would adopt a tool like EuterPen in their composition process with a hands-on user evaluation or, ideally, a longitudinal study performed in their work environment.

Beyond a more comprehensive evaluation, future work could also focus design efforts on leveraging structure even further. In our questionnaire, composers did react very positively to EuterPen's flexibility, they expressed much interest in features that benefit from structure as well – selection capabilities and the associated manipulations in particular. During their interview, one of the composers suggested pushing the direct manipulation of music symbols further, so that symbols would adapt in real-time while being moved around on the staff with the pen – for instance merging two symbols into a single (equivalent) one when they get juxtaposed. LA explained how he envisioned this: simulating the drag of an eighth rest with the pen and approaching another eighth rest, he said *"the eighth rest becomes a quarter rest [...] the software understands that you're moving the note. It's not just a graphical update, it's a rhythmic update."* Brainstorming about the same idea, DP suggested adding feedforward when dragging symbols this way: *"I think there needs to be a little barrier, a little help. Maybe it's just when you get to the middle of the two notes, that a little gray bar appears and you think 'Oh no, I'm going too far'."* Such advanced direct manipulations of notes and rests would require the tight coupling of direct manipulation interactions with structural transformations to re-write the engraved notation in real-time. This would require defining an entire rule-set of syntactically correct and semantically meaninfgul notation transformations that could be studied in future work.

Some interviewees also discussed with us the possibility to automatically interpret foreign material imported in EuterPen, such as an audio sample or a photo of a few bars quickly sketched on paper (both illustrated in Figure 4.9-b): *"What I'd like is for the photo taken there to be directly integrated into the score.*

That would be great" (AN); *"There must be a way for it to understand, 'Oh yes, this is music, let's try to play it.' [...] in fact, when you think about it, it's just music that's there, but in a different form"* (LA). While this is also left as a possible avenue for future work, recent advances in Artificial Intelligence might already have made this possible.

Chapter 5

Conclusion and Perspectives

As discussed in this manuscript, using pen and touch direct input on interactive surfaces raises many challenges with productivity tools and creativity support tools. The opportunities provided by the pen for precision and expressivity have been demonstrated in these contexts, but the means to articulate pen and structure had been overlooked. My thesis work focuses on this articulation with two use cases to answer the broad research question: *"How to design pen-based interactions for productivity and creativity on interactive surfaces?"* The present chapter summarizes my contributions toward answering this question and discusses possible future research directions to leverage these contributions.

5.1 Summary of Contributions

I started this thesis by noting that productivity and creativity require different approaches to favor an optimal workflow. Productivity relies on ability of a user to produce value efficiently (*i.e.*, efficiency), while creativity - which can be seen as creative productivity - additionally relies on the ability to produce novelty (*i.e.*, flexibility). Despite the benefits of digital pen, the structure of digital tools on interactive surfaces can be a barrier to achieving these goals and cause critical issues on four different levels: ($\mathbf{CI_1}$) limited display area for content, ($\mathbf{CI_2}$) limited number of features, ($\mathbf{CI_3}$) fatigue from gestures, ($\mathbf{CI_4}$) content-structure friction for interaction.

In order to mitigate these issues, I first investigated the challenges related to productivity with the specific research question RQ1:

> **RQ1**
>
> How can pen and structure be articulated in order to promote efficiency on interactive surfaces?

My first contribution, EunomInk (Chapter 3), proposes an answer to this question with a set of interaction techniques leveraging the pen to either **bypass the structure or embrace it**. Bypassing the structure allows it to be temporarily passive during an interaction to be able to act on the content without the structure being on the way. In my use case with spreadsheet programs, I demonstrated two possible forms this could take: breaking through the grid to directly access and manipulate data "under" it (*e.g.*, to input values, select and edit characters), or using widgets to avoid interacting with the grid altogether (*e.g.*, to pan the workspace or select ranges of rows and columns). However, embracing the structure remains more suited to transform the table on a global level (*e.g.*, to sort columns, delete rows, move groups of cells). This articulation also benefits from users' legacy knowledge of those programs on the desktop, while providing them with new ways to interact with their data. This design was informed by an analysis of current spreadsheet programs on interactive surfaces and an elicitation study in order to understand the interactions users *can do* and *could do*; it was then evaluated through a qualitative study that demonstrated its relevance.

Major parts of this contribution are drawn from a full paper published in TOCHI [30].

> **RQ2**
>
> How can pen and structure be articulated in order to promote both efficiency and flexibility on interactive surfaces?

My second contribution, EuterPen (Chapter 4), addresses this question by proposing a set of interaction techniques that mainly enable users to either **break the structure or embrace it**. Breaking the structure allows the user to locally set it aside while being able to quickly get it back. In my work with music notation programs, I showcased three possible forms this can take: breaking down musical elements to manipulate them directly without ambiguities (*e.g.*, to change their pitch, freely adjust their position, copy specific attributes), breaking the score's homogeneity (*e.g.*, to mix structured and unstructured content), and breaking the score's linearity (*e.g.*, to explore ideas in unconstrained canvases, edit the content of a measure without automatic adjustments). The structure, however, is never far away and can be easily retrieved. For instance, handwritten notation can be converted to engraved notation, and vice versa, at any time. Canvases can be readily hidden and do not disturb the layout of the score. This ensures a good stability of the content and allows the user to also benefit from the interactions embracing the structure, like selecting staves or measures. This articulation enables the user to adapt the system to their needs, while keeping the aesthetic and structural integrity of the score. This design was informed by nine interviews and a workshop with professional composers that helped me understand the creative process of composition; it was then evaluated through eight interviews in which composers confirmed its relevance.

Major parts of this contribution are drawn from a full paper published in CHI '24 [32] and from another full paper published in CHI '25 [31].

5.2 Discussion

Among my proposals to break the structure of music scores, some are very punctual while others are more prolonged in time. On the one hand, handwritten notation mixed with engraved notation has the potential to stay in the system for a long time, as the score can be saved in this state. On the other hand, breaking down a sequence of notes to copy its rhythmic pattern is an instantaneous action. The former implies a potentially durable departure from structure, while the latter implies a punctual deviation from its original form - as if nothing ever happened. This nuance suggests the need for a distinction between *breaking* the structure and only *bending* it.

Additionally, some features that leverage the pen's opportunities to bypass the grid structure in EunomInk, like the Minitable and Minivis plots, are also found useful in EuterPen to bypass the score's structure. For instance, the interactive minimap that gives an overview of the whole score and enables quick navigation, although a minor contribution from the set of interaction techniques, suggests that maybe more features tailored for efficiency alone may also benefit contexts where efficiency is not the only concern. *Ergo*, the combined goals of efficiency and flexibility might require more interactions that *bypass* the structure.

Similarly, the features that bend the music score by breaking down musical elements may also be useful with spreadsheets to break down data values and leverage their properties, suggesting yet again that the guidelines from Section 5.1 are not so clear-cut.

Some other contexts, however, might need a more binary approach. For instance, taking handwritten notes or painting on canvases are activities that are mainly structureless and require a complete separation from structure rather than attempts to compromise with it. In these cases, interactions should either *embrace* the structure (*e.g.*, to switch tools, format the document) or *disable* it entirely (*e.g.*, to write loose notes, draw freely).

These considerations motivate the proposal of a continuum of structural interaction in content editing programs, illustrated in Figure 5.1.

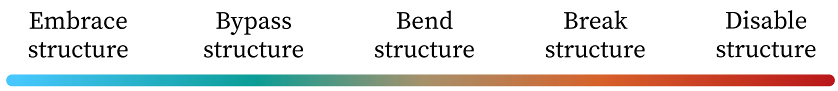


Figure 5.1: A continuum of structural interaction in content editing programs.

5.3 Key Insights for Future Work

The two investigations led in this thesis, complemented by the above considerations, enable us to circle back to the more global research question RQ0.

> **RQ0**
>
> How to design pen-based interactions to improve productivity and creativity on interactive surfaces?

We saw that designing pen-based interactions enabling the user to either embrace or bypass the structure can provide an efficient workflow; and that designing pen-based interactions enabling the user to either embrace, bypass, bend or break the structure has the potential to provide an efficient and flexible workflow. Doing so led to systems where the content exploited the whole display area (**CI$_1$**), where the user could access a wide range of features (**CI$_2$**), perform short and simple gestures (**CI$_3$**), with very few friction from interaction (**CI$_4$**). However, these insights are derived from the specific use cases of spreadsheet and music score editing, and I do not pretend that they offer general solutions that can be applied "as is" in other productivity and creativity contexts. Instead, I propose design steps that can be followed to support the development of optimal pen-based interactions in these contexts, emphasizing the importance of an adequate articulation with structure. These steps are as follows:

1. **Identify the Structure.** Structure simultaneously affords and constrains the space of actions a user can perform on a digital tool. Because of the specific characteristics of interactive surfaces and digital pen, identifying it is primordial to make it an ally rather than an obstacle. Sometimes the structure is explicit, as in the case of the grid in a spreadsheet program, visible in the entire workspace. But it can also be partly explicit and partly implicit, as in the case of music scores, or even mostly implicit in some other cases, as in the case of word processors. For this reason, the structure can be elicited from the use of these tools, and this step can be inverted with the next one.

2. **Collect the Needs.** The actions users engage in can be well-known and documented (*e.g.*, operations performed on spreadsheets), but can

also be more complex to grasp, especially in creative contexts. In these cases, it is crucial to step back and gather insights into users' needs — understanding why they use these tools or why they do not. Direct observations or interview sessions with experts, analyzed through a dual approach that examines both thought processes and workflows, can help in that regard. On top of creating a list of material to implement, this can shed light on sources of tension and help identify hidden structure.

3. **Classify the Actions.** Designers should determine which actions benefit from structure and which are hindered or prevented by it. The continuum in Figure 5.1 can serve as a basis for this delicate classification. Drawing insights from Chapter 3 and 4, designers might want to consider a structural interaction that leverages the structure for productivity (*e.g.*, from *embracing* to *bending* structure), and rather enables the user to adapt the system to their needs for creativity (*e.g.*, from *embracing* to *disabling* structure). Spreading the actions too much on this continuum might however lead to a confusing system, thus designers should focus on a few ways to interact with structure and only make occasional exceptions.

4. **Design the Interactions.** Instead of engaging in this step by wondering how pen and touch should be distributed across the tasks, designers can rather wonder how to approach the structural class the tasks fit in. For instance, tasks that *embrace* the structure might be suitable candidates for both pen and touch input, provided there is a clear division of labor [22, 76]. The precision of the pen makes it a good choice for tasks that need to *bypass* or *bend* the structure. Its expressivity can also have interesting uses to *break* the structure. Finally, tasks that rather *disable* the structure can be adressed with both pen and touch because of the absence of ambiguities. Designing interactions is a famously iterative activity, and designers might need to go back and forth between this step and the previous one in order to refine the classification of tasks.

5.4 Examples of Applications

This section explores five examples of productivity tools and creativity support tools that could benefit from the insights of this thesis and its proposal of a continuum of structural interaction. These WIMP tools are, for the moment, not adapted for direct input on interactive surfaces, but could be rethought to better articulate the digital pen with their inherent structure.

Presentation Programs: In famous presentation programs such as Microsoft Powerpoint[1], Google Slides[2], or Canva[3], the structure can be easily identified as twofold. There is the *slides tab*, displaying the list of slides as thumbnail-sized images, and the *presentation view*, enabling content editing on individual slides. Even if the designer chooses to save screen real estate by hiding the slides tab only to reveal them with a bezel gesture [76], this twofold structure seems essential to the purpose of this tool, which is to efficiently produce presentations by manipulating slides and their contents. The needs of users are rather straightforward and well-documented [176, 49]. A few examples are the need to create new slides, duplicate them, edit their content, and rearrange them. Creating new slides could be done by bypassing the structure, as it does not require to access the full list of slides. Duplicating slides could benefit from bending the structure: users often wants to keep only specific parts of the original slide, and could leverage the pen to break down its content and specify what needs to be duplicated. Editing the content of a slide might often need to bypass the structure in order to avoid the rules on layout and ease the selection of text or objects. Rearranging slides could rather embrace the structure, as users need to directly manipulate the full list of slides to make informed decisions. Users could however benefit from a widget like the Minitable 3 that bypasses the structure to quickly preview slides that are far away and drag them to a new position.

Digital Calendars: In digital calendars such as Apple Calendar[4] or Google Calendar[5], the structure is usually fourfold. The content (*e.g.*, the events) can be manipulated on the *day-grid*, the *week-grid*, the *month-grid* or the *year-grid*. Each one of these layers offers a different level of granularity and a unique way to interact with the content. Just like in presentation programs, the functionalities of these tools could leverage the structural interactions from the left side of the continuum. For instance, the creation of events - usually restricted to 15-minutes intervals - could benefit from bending the structure to create new events by the minute, thanks to the precision of the pen. The quick edition of events could also benefit from bending the structure to change the main parameters (*e.g.*, name, date, occurrences) without having to access the full list of parameters. However, this list would still be accessible with an interaction that embraces the structure, like a finger tap on the event. Embracing the structure could also be useful to copy paste or drag and drop events when no precision is needed. Finally, bypassing the structure could allow seamless

[1] https://www.microsoft.com/en/microsoft-365/?market=af
[2] https://workspace.google.com/products/slides/
[3] https://www.canva.com/
[4] https://www.icloud.com/calendar/
[5] https://calendar.google.com

navigation between the different grids and ease access to events that are hard to reach.

Raster Graphics Editors: Creativity support tools like Adobe Photoshop[6] and GIMP[7] have various uses from simple image editing to advanced artistic creations. Although their structure is partly explicit and based on superimposed layers, I would envision a study on users' thought and work processes to explore the diverse ways these tools are utilized and to uncover potential structural aspects that are not immediately apparent. After having derived a set of actions, designers could offer to break the structure to manipulate a layer's content away without risking to disturb previous work. The pen could bend the structure by breaking down the previous actions performed in an area, and choose which action to revert back to. Users could also disable the structure when committing to a brush for some time, and focus on drawing with the pen while touch input would be ignored to prevent accidental contacts.

3D Modeling Software: Programs such as Blender[8], Autodesk Maya[9], or Cinema 4D[10] are in fact a combination of several tools. They usually offer a pipeline to create 3D objects, add textures, place the objects in virtual scenes, set the lighting, render the animation, and refine it with post-processing. Arguably, each one of these tools possesses its own distinct structure, and the need for formative studies with experts is paramount in order to understand their respective challenges. Nonetheless, such programs could greatly benefit from a well-defined structural interaction with the pen. First of all, while users can draw in 2D with the pen, a combination of pen and touch embracing the structure of the 3D environment could allow users to move along the third axis and draw in 3D. Breaking the structure could be useful for the opposite goal: taking an object from the 3D environment and refine it in a space away from its constraints (*e.g.*, position, temporality, lighting) like the pensieves in Chapter 4. The pen could finally bend the structure to reach small objects hidden behind bigger ones, or in dense settings.

Word Processors: Programs like Microsoft Word[11] or Google Docs[12] are used to write and format text. However, they can be used in both productive contexts (*e.g.*, to write reports) and creative contexts (*e.g.*, to write books,

[6] https://www.adobe.com/products/photoshop.html
[7] https://www.gimp.org/
[8] https://www.blender.org/
[9] https://www.autodesk.com/products/maya/
[10] https://www.maxon.net/en/product-detail/cinema-4d
[11] https://www.microsoft.com/en-us/microsoft-365/word
[12] https://workspace.google.com/products/docs/

poetry) and should ideally be able to support both. Their structure is mainly implicit, as the content is displayed in a linear fashion. However, there are a lot of hidden rules like automatic adjustments, styles, and formatting that can be seen as a structure. In order to help users be efficient, the pen could bypass the structure to avoid the rules on layout and select text intuitively, like it did with the subcell selections in Chapter 3. In order to help them be flexible, the pen could break the structure to manipulate new handwritten content and previously written one alike. It could also bend the structure by allowing users to see alternatives to the content they are writing - like synonyms or whole sentences - integrated in their content.

5.5 Final Word

In this dissertation, I have looked into the relationship between pen-based interactions and the constraints inherent to productivity and creativity tools. While the pen offers unique advantages of precision and expressivity on interactive surfaces, its coupling with underlying structures has not been adequately investigated. Through the discussion of two specific use cases—spreadsheet and music score editing—I have advocated that having the capability to embrace or bypass the structure can lead to productive workflows. Allowing users to play with the structure in more malleable ways, such as bending or breaking it, can also promote much-needed flexibility.

Although these findings are based on particular domains, they highlight the potential of pen-based interactions to benefit from the advantages of structure and pen affordances. By proposing a range of structural interactions and a design process, I hope to make it easier for designers to consider the subtle ways in which pen input can be leveraged for optimal user experience.

Bibliography

[1] Cyril J. Adcock and William A. Martin. “Flexibility and Creativity”. In: *The Journal of General Psychology* 85.1 (1971). PMID: 5566736, pp. 71–76. doi: 10.1080/00221309.1971.9920654. eprint: https://doi.org/10.1080/00221309.1971.9920654. url: https://doi.org/10.1080/00221309.1971.9920654.

[2] Anand Agarawala and Ravin Balakrishnan. “Keepin’it real: pushing the desktop metaphor with physics, piles and the pen”. In: *Proceedings of the SIGCHI conference on Human Factors in computing systems*. 2006, pp. 1283–1292.

[3] Michelle Annett, Fraser Anderson, Walter F. Bischof, and Anoop Gupta. “The Pen is Mightier: Understanding Stylus Behaviour While Inking on Tablets”. In: *Proceedings of Graphics Interface 2014*. GI ’14. Montreal, Quebec, Canada: Canadian Information Processing Society, 2014, pp. 193–200. isbn: 9781482260038.

[4] Michelle Annett, Anoop Gupta, and Walter F. Bischof. “Exploring and Understanding Unintended Touch during Direct Pen Interaction”. In: *ACM Trans. Comput.-Hum. Interact.* 21.5 (Nov. 2014). issn: 1073-0516. doi: 10.1145/2674915. url: https://doi.org/10.1145/2674915.

[5] J. Anstice, T. Bell, A. Cockburn, and M. Setchell. “The design of a pen-based musical input system”. In: *Proceedings Sixth Australian Conference on Computer-Human Interaction*. OzCHI. 1996, pp. 260–267. doi: 10.1109/OZCHI.1996.560019.

[6] Caroline Appert and Shumin Zhai. “Using Strokes as Command Shortcuts: Cognitive Benefits and Toolkit Support”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’09. Boston, MA, USA: Association for Computing Machinery, 2009, pp. 2289–2298. isbn: 9781605582467. doi: 10.1145/1518701.1519052. url: https://doi.org/10.1145/1518701.1519052.

[7] Apple. *iPad Pro 11-inch (1st generation) - Technical Specifications*. https://support.apple.com/en-us/111974. Last accessed: 2024-12-15. 2024.

[8] Ian Arawjo, Anthony DeArmas, Michael Roberts, Shrutarshi Basu, and Tapan Parikh. "Notational Programming for Notebook Environments: A Case Study with Quantum Circuits". In: *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. UIST '22. Bend, OR, USA: Association for Computing Machinery, 2022. isbn: 9781450393201. doi: 10.1145/3526113.3545619. url: https://doi.org/10.1145/3526113.3545619.

[9] Benjamin Bach, Natalie Kerracher, Kyle Wm. Hall, Sheelagh Carpendale, Jessie Kennedy, and Nathalie Henry Riche. "Telling Stories about Dynamic Networks with Graph Comics". In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. CHI '16. San Jose, California, USA: Association for Computing Machinery, 2016, pp. 3670–3682. isbn: 9781450333627. doi: 10.1145/2858036.2858387. url: https://doi.org/10.1145/2858036.2858387.

[10] Lyn Bartram, Michael Correll, and Melanie Tory. "Untidy Data: The Unreasonable Effectiveness of Tables". In: *IEEE Transactions on Visualization and Computer Graphics* 28.1 (2022), pp. 686–696. doi: 10.1109/TVCG.2021.3114830.

[11] Olivier Bau and Wendy E. Mackay. "OctoPocus: a dynamic guide for learning gesture-based command sets". In: *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology*. UIST '08. Monterey, CA, USA: Association for Computing Machinery, 2008, pp. 37–46. isbn: 9781595939753. doi: 10.1145/1449715.1449724. url: https://doi.org/10.1145/1449715.1449724.

[12] 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*. CHI '00. The Hague, The Netherlands: Association for Computing Machinery, 2000, pp. 446–453. isbn: 1581132166. doi: 10.1145/332040.332473. url: https://doi.org/10.1145/332040.332473.

[13] Amelia Bellamy-Roayds, Bogdan Brinza, Chris Lilley, Dirk Schulze, David Storey, and Eric Willigers. *Scalable Vector Graphics (SVG) 2 - W3C Candidate Recommendation*. https://www.w3.org/TR/SVG2/. Oct. 2018.

[14] Stan Bennett. "The Process of Musical Creation: Interviews with Eight Composers". In: *Journal of Research in Music Education* 24.1 (1976), pp. 3–13. (Visited on 08/30/2023).

[15] A. F. Blackwell, C. Britton, A. Cox, T. R. G. Green, C. Gurr, G. Kadoda, M. S. Kutar, M. Loomes, C. L. Nehaniv, M. Petre, C. Roast, C. Roe, A. Wong, and R. M. Young. "Cognitive Dimensions of Notations: Design Tools for Cognitive Technology". In: *Cognitive Technology: Instruments of Mind*. Ed. by Meurig Beynon, Chrystopher L. Nehaniv, and Kerstin Dautenhahn. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 325–341. isbn: 978-3-540-44617-0.

[16] Alan F Blackwell, Thomas RG Green, and Douglas JE Nunn. "Cognitive dimensions and musical notation systems". In: *Workshop on Notation and Music Information Retrieval in the Computer Age at the International Computer Music Conference*. 2000.

[17] Alan F. Blackwell and Thomas R. G. Green. "Investment of attention as an analytic approach to cognitive dimensions". In: *Annual Workshop of the Psychology of Programming Interest Group*. 1999. url: https://api.semanticscholar.org/CorpusID:47018198.

[18] Joshua Blake. "Natural user interfaces". In: *Net. 1st edn. New York* (2011).

[19] Ann Blandford. "Semi-structured qualitative studies". In: *The Encyclopedia of Human-Computer Interaction, 2nd edition*. Ed. by Mads Soegaard and Rikke Friis Dam. Interaction Design Foundation, 2013.

[20] Margaret A. Boden. "Computer Models of Creativity". In: *AI Mag.* 30.3 (Sept. 2009), pp. 23–34. issn: 0738-4602. doi: 10.1609/aimag.v30i3.2254. url: https://doi.org/10.1609/aimag.v30i3.2254.

[21] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. "D3: Data-Driven Documents". In: *IEEE Transactions on Visualization and Computer Graphics* 17.12 (2011), pp. 2301–2309. doi: 10.1109/TVCG.2011.185.

[22] Peter Brandl, Clifton Forlines, Daniel Wigdor, Michael Haller, and Chia Shen. "Combining and Measuring the Benefits of Bimanual Pen and Direct-Touch Interaction on Horizontal Interfaces". In: *Proceedings of the Working Conference on Advanced Visual Interfaces*. AVI '08. Napoli, Italy: Association for Computing Machinery, 2008, pp. 154–161. isbn: 9781605581415. doi: 10.1145/1385569.1385595. url: https://doi.org/10.1145/1385569.1385595.

[23] Jean Bresson, Carlos Agon, and Gérard Assayag. "OpenMusic: Visual Programming Environment for Music Composition, Analysis and Research". In: *Proceedings of the 19th ACM International Conference on Multimedia*. MM '11. Scottsdale, Arizona, USA: Association for Computing Machinery, 2011, pp. 743–746. isbn: 9781450306164. doi: 10.1145/2072298.2072434. url: https://doi.org/10.1145/2072298.2072434.

[24] Britannica. *Definition of Creativity*. https://www.britannica.com/topic/creativity. Last accessed: 2024-10-27. 2024.

[25] Jeffrey Browne, Bongshin Lee, Sheelagh Carpendale, Nathalie Riche, and Timothy Sherwood. "Data Analysis on Interactive Whiteboards through Sketch-Based Interaction". In: *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*. ITS '11. Kobe, Japan: Association for Computing Machinery, 2011, pp. 154–157. isbn: 9781450308717. doi: 10.1145/2076354.2076383. url: https://doi.org/10.1145/2076354.2076383.

[26] Matt Brubeck, Rick Byers, Patrick H. Lauke, and Navid Zolghadr. *Pointer Events Level 2 - W3C Recommendation*. https://www.w3.org/TR/pointerevents2/. Apr. 2019.

[27] Jorge Calvo-Zaragoza, Jan Hajič Jr., and Alexander Pacha. "Understanding Optical Music Recognition". In: *ACM Comput. Surv.* 53.4 (July 2020). issn: 0360-0300. doi: 10.1145/3397499. url: https://doi.org/10.1145/3397499.

[28] Drini Cami, Fabrice Matulic, Richard G. Calland, Brian Vogel, and Daniel Vogel. "Unimanual Pen+Touch Input Using Variations of Precision Grip Postures". In: *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. UIST '18. Berlin, Germany: Association for Computing Machinery, 2018, pp. 825–837. isbn: 9781450359481. doi: 10.1145/3242587.3242652. url: https://doi.org/10.1145/3242587.3242652.

[29] John M. Carroll and Judith Reitman Olson. "Chapter 2 - Mental Models in Human-Computer Interaction". In: *Handbook of Human-Computer Interaction*. Ed. by MARTIN HELANDER. Amsterdam: North-Holland, 1988, pp. 45–65. isbn: 978-0-444-70536-5. doi: https://doi.org/10.1016/B978-0-444-70536-5.50007-5. url: https://www.sciencedirect.com/science/article/pii/B9780444705365500075.

[30] Vincent Cavez, Caroline Appert, and Emmanuel Pietriga. “Spreadsheets on Interactive Surfaces: Breaking through the Grid with the Pen”. In: *ACM Trans. Comput.-Hum. Interact.* 31.2 (Jan. 2024). issn: 1073-0516. doi: 10.1145/3630097. url: https://doi.org/10.1145/3630097.

[31] Vincent Cavez, Catherine Letondal, Caroline Appert, and Emmanuel Pietriga. “EuterPen: Unleashing Creative Expression in Music Score Writing”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. CHI ’25. New York, NY, USA: Association for Computing Machinery, 2025. doi: 10.1145/3706598.3713488. url: https://doi.org/10.1145/3706598.3713488.

[32] Vincent Cavez, Catherine Letondal, Emmanuel Pietriga, and Caroline Appert. “Challenges of Music Score Writing and the Potentials of Interactive Surfaces”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. CHI ’24. New York, NY, USA: Association for Computing Machinery, 2024. isbn: 9798400703300. doi: 10.1145/3613904.3642079. url: https://doi.org/10.1145/3613904.3642079.

[33] George Chalhoub and Advait Sarkar. “"It’s Freedom to Put Things Where My Mind Wants”: Understanding and Improving the User Experience of Structuring Data in Spreadsheets”. 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: 10.1145/3491102.3501833. url: https://doi.org/10.1145/3491102.3501833.

[34] Robert G. Chambers. *Journal of Productivity Analysis*. https://link.springer.com/journal/11123. Last accessed: 2024-11-24. 2024.

[35] Ran Chen, Di Weng, Yanwei Huang, Xinhuan Shu, Jiayi Zhou, Guodao Sun, and Yingcai Wu. “Rigel: Transforming Tabular Data by Declarative Mapping”. In: *IEEE Transactions on Visualization and Computer Graphics* 29.1 (2023), pp. 128–138. doi: 10.1109/TVCG.2022.3209385.

[36] Mohit Muthanna Cheppudira. *VexFlow - JavaScript Music Notation and Guitar Tablature*. https://www.vexflow.com. Last accessed: 2024-08-01. 2024.

[37] Andy Cockburn, Amy Karlson, and Benjamin B. Bederson. "A Review of Overview+detail, Zooming, and Focus+context Interfaces". In: *ACM Comput. Surv.* 41.1 (Jan. 2009). issn: 0360-0300. doi: 10.1145/1456650.1456652. url: https://doi.org/10.1145/1456650.1456652.

[38] David Collins. "A synthesis process model of creative thinking in music composition". In: *Psychology of Music* 33.2 (2005), pp. 193–216. doi: 10.1177/0305735605050651.

[39] Tim Coughlan and Peter Johnson. "Interaction in Creative Tasks". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '06. Montréal, Québec, Canada: Association for Computing Machinery, 2006, pp. 531–540. isbn: 1595933727. doi: 10.1145/1124772.1124854. url: https://doi.org/10.1145/1124772.1124854.

[40] Andrew Crotty, Alex Galakatos, Emanuel Zgraggen, Carsten Binnig, and Tim Kraska. "Vizdom: Interactive Analytics through Pen and Touch". In: *Proc. VLDB Endow.* 8.12 (Aug. 2015), pp. 2024–2027. issn: 2150-8097. doi: 10.14778/2824032.2824127. url: https://doi.org/10.14778/2824032.2824127.

[41] Andries van Dam. "Post-WIMP user interfaces". In: *Commun. ACM* 40.2 (Feb. 1997), pp. 63–67. issn: 0001-0782. doi: 10.1145/253671.253708. url: https://doi.org/10.1145/253671.253708.

[42] Philip L Davidson and Jefferson Y Han. "Extending 2D object arrangement with pressure-sensitive layering cues". In: *Proceedings of the 21st annual ACM symposium on User interface software and technology*. 2008, pp. 87–90.

[43] Cambridge Academic Content Dictionary. *Definition of Creativity*. https://dictionary.cambridge.org/us/dictionary/english/creativity. Last accessed: 2024-10-27. 2024.

[44] Dorico. *Music Notation and Composition Software for macOS, Windows and iPad*. https://www.steinberg.net/dorico/. Last accessed: 2024-12-15. 2024.

[45] Paul Dourish. "Spreadsheets and Spreadsheet Events in Organizational Life". In: *The Stuff of Bits: An Essay on the Materialities of Information*. The MIT Press, May 2017. isbn: 9780262340120. doi: 10.7551/mitpress/10999.003.0005. url: https://doi.org/10.7551/mitpress/10999.003.0005.

[46] Lisa A. Elkin, Matthew Kay, James J. Higgins, and Jacob O. Wobbrock. “An Aligned Rank Transform Procedure for Multifactor Contrast Tests”. In: *Proceedings of the 34th Annual ACM Symposium on User Interface Software and Technology*. UIST ’21. Virtual: Association for Computing Machinery, 2021, pp. 754–768. doi: 10.1145/3472749.3474784. url: https://doi.org/10.1145/3472749.3474784.

[47] ExplainingComputers. *Classic PC: Newton MessagePad 110*. https://youtu.be/89XnXe3FnCo?si=IWWeYjRFvOST8Kmh. Last accessed: 2024-12-15. 2024.

[48] Farzin Farhadi-Niaki, S. Ali Etemad, and Ali Arya. “Design and Usability Analysis of Gesture-Based Control for Common Desktop Tasks”. In: *Human-Computer Interaction. Interaction Modalities and Techniques*. Ed. by Masaaki Kurosu. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 215–224. isbn: 978-3-642-39330-3.

[49] David Farkas. “Understanding and Using PowerPoint”. In: (Jan. 2005).

[50] Göran Folkestad, David J. Hargreaves, and Berner Lindström. “Compositional Strategies in Computer-Based Music-Making”. In: *British Journal of Music Education* 15.1 (1998), pp. 83–97. doi: 10.1017/S0265051700003788.

[51] E.H.P. Frankema. *Kantoorinnovatie in economisch perspectief*. Dutch. Center for People and Buildings, 2003.

[52] Jason Freeman. “Graph theory: interfacing audiences into the compositional process”. In: *Proceedings of the 7th International Conference on New Interfaces for Musical Expression*. NIME ’07. New York, New York: Association for Computing Machinery, 2007, pp. 260–263. isbn: 9781450378376. doi: 10.1145/1279740.1279794. url: https://doi.org/10.1145/1279740.1279794.

[53] Mathias Frisch, Jens Heydekorn, and Raimund Dachselt. “Diagram Editing on Interactive Displays Using Multi-touch and Pen Gestures”. In: *Diagrammatic Representation and Inference*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 182–196.

[54] Mathias Frisch, Ricardo Langner, and Raimund Dachselt. “Neat: A Set of Flexible Tools and Gestures for Layout Tasks on Interactive Displays”. In: *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*. ITS ’11. Kobe, Japan: Association for Computing Machinery, 2011, pp. 1–10.

isbn: 9781450308717. doi: 10 . 1145 / 2076354 . 2076356. url: https://doi.org/10.1145/2076354.2076356.

[55] Jérémie Garcia, Louis Bigo, Antoine Spicher, and Wendy E. Mackay. "PaperTonnetz: supporting music composition with interactive paper". In: *CHI '13 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '13. Paris, France: Association for Computing Machinery, 2013, pp. 3051–3054. isbn: 9781450319522. doi: 10 . 1145 / 2468356 . 2479608. url: https://doi.org/10.1145/2468356.2479608.

[56] Jérémie Garcia, Theophanis Tsandilas, Carlos Agon, and Wendy Mackay. "InkSplorer: Exploring Musical Ideas on Paper and Computer". In: *New Interfaces for Musical Expression*. NIME '11. Oslo, Norway, May 2011.

[57] 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, pp. 1825–1828. isbn: 9781450310154. doi: 10 . 1145 / 2207676 . 2208316. url: https://doi.org/10.1145/2207676.2208316.

[58] Jérémie Garcia, Theophanis Tsandilas, Carlos Agon, and Wendy Mackay. "PaperComposer: creating interactive paper interfaces for music composition". In: *Proceedings of the 26th Conference on l'Interaction Homme-Machine*. IHM '14. Villeneuve d'Ascq, France: Association for Computing Machinery, 2014, pp. 1–8. isbn: 9781450329354. doi: 10 . 1145 / 2670444 . 2670450. url: https://doi.org/10.1145/2670444.2670450.

[59] Jérémie Garcia, Theophanis Tsandilas, Carlos Agon, and Wendy E. Mackay. "Structured Observation with Polyphony: A Multifaceted Tool for Studying Music Composition". In: *Proceedings of the 2014 Conference on Designing Interactive Systems*. DIS '14. Vancouver, BC, Canada: Association for Computing Machinery, 2014, pp. 199–208. isbn: 9781450329026. doi: 10.1145/2598510.2598512. url: https://doi.org/10.1145/2598510.2598512.

[60] Travis Gesslein, Verena Biener, Philipp Gagel, Daniel Schneider, Per Ola Kristensson, Eyal Ofek, Michel Pahud, and Jens Grubert. "Pen-based Interaction with Spreadsheets in Mobile Virtual Reality". In: *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2020, pp. 361–373. doi: 10.1109/ISMAR50242.2020.00063.

[61] James J Gibson. *The Theory of Affordances. The Ecological Approach to Visual Perception*. Houghton Mifflin, 1979.

[62] Eric J. Golin and Steven P. Reiss. "The specification of visual language syntax". In: *Journal of Visual Languages & Computing (JVLC)* 1.2 (1990), pp. 141–157. issn: 1045-926X. doi: https://doi.org/10.1016/S1045-926X(05)80013-8.

[63] T. R. G. Green and M. Petre. "Usability Analysis of Visual Programming Environments: A 'Cognitive Dimensions' Framework". In: *Journal of Visual Languages & Computing* 7.2 (1996), pp. 131–174. issn: 1045-926X. doi: https://doi.org/10.1006/jvlc.1996.0009. url: https://www.sciencedirect.com/science/article/pii/S1045926X96900099.

[64] Thomas Green and Alan Blackwell. "Cognitive dimensions of information artefacts: a tutorial". In: *BCS-HCI Conference*. Vol. 98. 1998, pp. 1–75.

[65] Paul Grünbacher, Rudolf Hanl, and Lukas Linsbauer. "Using Music Features for Managing Revisions and Variants of Musical Scores". In: *Computer Music Journal* (Oct. 2024), pp. 1–44. doi: 10.1162/comj_a_00691.

[66] GSMArena. *BlackBerry Bold 9000*. https://www.gsmarena.com/blackberry_bold_9000-2370.php. Last accessed: 2024-12-15. 2024.

[67] Yves Guiard. "Asymmetric Division of Labor in Human Skilled Bimanual Action". In: *Journal of Motor Behavior* 19.4 (Dec. 1987), pp. 486–517. issn: 0022-2895. doi: 10.1080/00222895.1987.10735426. url: https://doi.org/10.1080/00222895.1987.10735426.

[68] J. P. Guilford. "Creativity." In: *American Psychologist* 5 (1950), pp. 444–454. doi: 10.1037/h0063487.

[69] Sumit Gulwani. "Automating String Processing in Spreadsheets Using Input-Output Examples". In: *Proceedings of the 38th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*. POPL '11. Austin, Texas, USA: Association for Computing Machinery, 2011, pp. 317–330. isbn: 9781450304900. doi: 10.1145/1926385.1926423. url: https://doi.org/10.1145/1926385.1926423.

[70] Sumit Gulwani, William R. Harris, and Rishabh Singh. "Spreadsheet Data Manipulation Using Examples". In: *Commun. ACM* 55.8 (Aug. 2012), pp. 97–105. issn: 0001-0782. doi: 10.1145/2240236.2240260. url: https://doi.org/10.1145/2240236.2240260.

[71] Philip J. Guo, Sean Kandel, Joseph M. Hellerstein, and Jeffrey Heer. "Proactive Wrangling: Mixed-Initiative End-User Programming of Data Transformation Scripts". In: *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*. UIST '11. Santa Barbara, California, USA: Association for Computing Machinery, 2011, pp. 65–74. isbn: 9781450307161. doi: 10.1145/2047196.2047205. url: https://doi.org/10.1145/2047196.2047205.

[72] Lone Koefoed Hansen and Peter Dalsgaard. "Note to self: stop calling interfaces "natural"". In: *Proceedings of The Fifth Decennial Aarhus Conference on Critical Alternatives*. 2015, pp. 65–68.

[73] William R. Harris and Sumit Gulwani. "Spreadsheet Table Transformations from Examples". In: *SIGPLAN Not.* 46.6 (June 2011), pp. 317–328. issn: 0362-1340. doi: 10.1145/1993316.1993536. url: https://doi.org/10.1145/1993316.1993536.

[74] D.L. Hecht. "Printed embedded data graphical user interfaces". In: *IEEE Computer* 34.3 (2001), pp. 47–55. doi: 10.1109/2.910893.

[75] Ken Hinckley, Xiaojun Bi, Michel Pahud, and Bill Buxton. "Informal Information Gathering Techniques for Active Reading". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '12. Austin, Texas, USA: Association for Computing Machinery, 2012, pp. 1893–1896. isbn: 9781450310154. doi: 10.1145/2207676.2208327. url: https://doi.org/10.1145/2207676.2208327.

[76] Ken Hinckley, Koji Yatani, Michel Pahud, Nicole Coddington, Jenny Rodenhouse, Andy Wilson, Hrvoje Benko, and Bill Buxton. "Pen + Touch = New Tools". In: *Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology*. UIST '10. New York, New York, USA: Association for Computing Machinery, 2010, pp. 27–36. isbn: 9781450302715. doi: 10.1145/1866029.1866036. url: https://doi.org/10.1145/1866029.1866036.

[77] Jane Hoffswell and Zhicheng Liu. "Interactive Repair of Tables Extracted from PDF Documents on Mobile Devices". In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19. Glasgow, Scotland Uk: Association for Computing Machinery, 2019, pp. 1–13. isbn: 9781450359702. doi: 10.1145/3290605.3300523. url: https://doi.org/10.1145/3290605.3300523.

[78] Karen Holtzblatt and Hugh Beyer. *Contextual design: defining customer-centered systems*. Elsevier, 1997.

[79] Eva Hornecker. "Beyond affordance: tangibles' hybrid nature". In: *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction*. 2012, pp. 175–182.

[80] Andy Hunt and Marcelo Wanderley. "Mapping performer parameters to synthesis engines". In: *Organised Sound* 7 (Aug. 2002), pp. 97–108. doi: 10.1017/S1355771802002030.

[81] T. Igarashi, J.D. Mackinlay, Bay-Wei Chang, and P.T. Zellweger. "Fluid visualization of spreadsheet structures". In: *Proceedings. 1998 IEEE Symposium on Visual Languages*. 1998, pp. 118–125. doi: 10.1109/VL.1998.706154.

[82] W. Isaacson. *Steve Jobs*. Essais et documents. JC Lattès, 2011. isbn: 9782709638821. url: https://books.google.fr/books?id=swsZuHxsJDwC.

[83] Will Cukierski Jessica Li. *Titanic - Machine Learning from Disaster*. 2012. url: https://kaggle.com/competitions/titanic.

[84] Zhongjun Jin, Michael R. Anderson, Michael Cafarella, and H. V. Jagadish. "Foofah: Transforming Data By Example". In: *Proceedings of the 2017 ACM International Conference on Management of Data*. SIGMOD '17. Chicago, Illinois, USA: Association for Computing Machinery, 2017, pp. 683–698. isbn: 9781450341974. doi: 10.1145/3035918.3064034. url: https://doi.org/10.1145/3035918.3064034.

[85] Jaemin Jo, Sehi L'Yi, Bongshin Lee, and Jinwook Seo. "TouchPivot: Blending WIMP & Post-WIMP Interfaces for Data Exploration on Tablet Devices". In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI '17. Denver, Colorado, USA: Association for Computing Machinery, 2017, pp. 2660–2671. isbn: 9781450346559. doi: 10.1145/3025453.3025752. url: https://doi.org/10.1145/3025453.3025752.

[86] Sean Kandel, Andreas Paepcke, Joseph Hellerstein, and Jeffrey Heer. "Wrangler: Interactive Visual Specification of Data Transformation Scripts". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '11. Vancouver, BC, Canada: Association for Computing Machinery, 2011, pp. 3363–3372. isbn: 9781450302289. doi: 10.1145/1978942.1979444. url: https://doi.org/10.1145/1978942.1979444.

[87] Audrey Kaplan and Stanley J. Aronoff. "Total Workplace Performance: Rethinking The Office Environment". In: 1995. url: https://api.semanticscholar.org/CorpusID:132250447.

[88] Jun Kato, Kenta Hara, and Nao Hirasawa. "Griffith: A Storyboarding Tool Designed with Japanese Animation Professionals". In: *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. CHI '24. Honolulu, HI, USA: Association for Computing Machinery, 2024. isbn: 9798400703300. doi: 10.1145/3613904.3642121. url: https://doi.org/10.1145/3613904.3642121.

[89] David Kirsh. "Thinking with external representations". In: *AI & Society* 25.4 (2010), pp. 441–454. doi: 10.1007/s00146-010-0272-8.

[90] Gordon Kurtenbach and William Buxton. "Issues in combining marking and direct manipulation techniques". In: *Proceedings of the 4th annual ACM symposium on User interface software and technology*. 1991, pp. 137–144.

[91] Sam Lau, Sruti Srinivasa Srinivasa Ragavan, Ken Milne, Titus Barik, and Advait Sarkar. "TweakIt: Supporting End-User Programmers Who Transmogrify Code". In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. CHI '21. Yokohama, Japan: Association for Computing Machinery, 2021. isbn: 9781450380966. doi: 10.1145/3411764.3445265. url: https://doi.org/10.1145/3411764.3445265.

[92] Mikael Laurson and Mika Kuuskankare. "Towards Idiomatic and Flexible Score-based Gestural Control with a Scripting Language". In: *Proceedings of the International Conference on New Interfaces for Musical Expression*. Genoa, Italy, 2008, pp. 34–37. doi: 10.5281/zenodo.1179589. url: http://www.nime.org/proceedings/2008/nime2008_034.pdf.

[93] Bongshin Lee, Greg Smith, Nathalie Henry Riche, Amy Karlson, and Sheelagh Carpendale. "SketchInsight: Natural data exploration on interactive whiteboards leveraging pen and touch interaction". In: *IEEE Pacific Visualization Symposium (PacificVis)*. 2015, pp. 199–206. doi: 10.1109/PACIFICVIS.2015.7156378.

[94] Bongshin Lee, Arjun Srinivasan, Petra Isenberg, and John Stasko. "Post-WIMP Interaction for Information Visualization". In: *Foundations and Trends in Human–Computer Interaction* 14.1 (2021), pp. 1–95. issn: 1551-3955. doi: 10.1561/1100000081. url: http://dx.doi.org/10.1561/1100000081.

[95] Annick Leroy, Giovanni Müller, and Guy E. Garnett. "The Design of a Pen-Based Music Notation System". In: *Proceedings of the 1994 International Computer Music Conference*. ICMC '94. 1994.

[96] Catherine Letondal and Wendy Mackay. "The Paperoles Project: An Analysis of Paper Use by Music Composers". In: *Proceedings of the 2nd International Workshop on Collaborating over Paper and Digital Documents*. CoPADD. 2007, pp. 27–30.

[97] Weiyuan Liu. "Natural user interface-next mainstream product user interface". In: *2010 IEEE 11th International Conference on Computer-Aided Industrial Design & Conceptual Design 1*. Vol. 1. IEEE. 2010, pp. 203–205.

[98] Claudio Longobardi, Rocco Quaglia, and Nathalie O Iotti. "Reconsidering the scribbling stage of drawing: a new perspective on toddlers' representational processes". In: *Frontiers in Psychology* 6 (2015), p. 1227.

[99] Ewa Luger and Abigail Sellen. "" Like Having a Really Bad PA" The Gulf between User Expectation and Experience of Conversational Agents". In: *Proceedings of the 2016 CHI conference on human factors in computing systems*. 2016, pp. 5286–5297.

[100] Guy Lüthi, Andreas Rene Fender, and Christian Holz. "DeltaPen: A Device with Integrated High-Precision Translation and Rotation Sensing on Passive Surfaces". In: *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. UIST '22. Bend, OR, USA: Association for Computing Machinery, 2022. isbn: 9781450393201. doi: 10.1145/3526113.3545655. url: https://doi.org/10.1145/3526113.3545655.

[101] Anna Macaranas, Alissa N Antle, and Bernhard E Riecke. "What is intuitive interaction? Balancing users' performance and satisfaction with natural user interfaces". In: *Interacting with Computers* 27.3 (2015), pp. 357–370.

[102] Wendy Mackay and Visiting Professor. "Responding to cognitive overload: Co-adaptation between users and technology". In: *Intellectica* 30 (July 2000). doi: 10.3406/intel.2000.1597.

[103] Alessio Malizia and Andrea Bellucci. "The artificiality of natural user interfaces". In: *Communications of the ACM* 55.3 (2012), pp. 36–38.

[104] Joseph Malloch, David Birnbaum, Elliot Sinyor, and Marcelo Wanderley. "Towards a new conceptual framework for digital musical instruments". In: (Jan. 2006).

[105] Joseph Malloch, Jérémie Garcia, Marcelo M Wanderley, Wendy E Mackay, Michel Beaudouin-Lafon, and Stéphane Huot. "A design WorkBench for interactive music systems". In: *New Directions in Music and Human-Computer Interaction* (2019), pp. 23–40.

[106] Daniel Manesh and Eran Egozy. "Exquisite Score: A System for Collaborative Musical Composition". In: *Proceedings of the International Conference on New Interfaces for Musical Expression*. Copenhagen, Denmark: Aalborg University Copenhagen, 2017, pp. 93–98. doi: 10.5281/zenodo.1176183. url: http://www.nime.org/proceedings/2017/nime2017_paper0019.pdf.

[107] Catherine C. Marshall. "Annotation: From Paper Books to the Digital Library". In: *Proc. of the ACM Int. Conf. on Digital libraries*. ACM Press, July 1997, pp. 131–140. doi: 10.1145/263690.263806.

[108] Raul Masu, Nuno N. Correia, and Teresa Romao. "NIME Scores: a Systematic Review of How Scores Have Shaped Performance Ecologies in NIME". In: *NIME 2021*. https://nime.pubpub.org/pub/41cj1pyt. Apr. 2021.

[109] Fabrice Matulic, Riku Arakawa, Brian Vogel, and Daniel Vogel. "PenSight: Enhanced Interaction with a Pen-Top Camera". In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI '20. Honolulu, HI, USA: Association for Computing Machinery, 2020, pp. 1–14. isbn: 9781450367080. doi: 10.1145/3313831.3376147. url: https://doi.org/10.1145/3313831.3376147.

[110] Fabrice Matulic and Moira C. Norrie. "Pen and Touch Gestural Environment for Document Editing on Interactive Tabletops". In: *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces*. ITS '13. St. Andrews, Scotland, United Kingdom: Association for Computing Machinery, 2013, pp. 41–50. isbn: 9781450322713. doi: 10.1145/2512349.2512802. url: https://doi.org/10.1145/2512349.2512802.

[111] Fabrice Matulic and Moira C. Norrie. "Supporting Active Reading on Pen and Touch-Operated Tabletops". In: *Proceedings of the International Working Conference on Advanced Visual Interfaces*. AVI '12. Capri Island, Italy: Association for Computing Machinery, 2012, pp. 612–619. isbn: 9781450312875. doi: 10.1145/2254556.2254669. url: https://doi.org/10.1145/2254556.2254669.

[112] Fabrice Matulic, Daniel Vogel, and Raimund Dachselt. "Hand Contact Shape Recognition for Posture-Based Tabletop Widgets and Interaction". In: *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces*. ISS '17. Brighton, United Kingdom: Association for Computing Machinery, 2017, pp. 3–11. isbn: 9781450346917. doi: 10.1145/3132272.3134126. url: https://doi.org/10.1145/3132272.3134126.

[113] Sarnoff Mednick. "The associative basis of the creative process." In: *Psychological review* 69.3 (1962), p. 220.

[114] Hrim Mehta, Adam James Bradley, Mark Hancock, and Christopher Collins. "Metatation: Annotation as Implicit Interaction to Bridge Close and Distant Reading". In: *ACM Trans. on Computer-Human Interaction (TOCHI)* (2017), 35:1–35:41. doi: 10.1145/3131609.

[115] Merriam-Webster. *Definition of Creativity*. https://www.merriam-webster.com/dictionary/creativity. Last accessed: 2024-10-27. 2024.

[116] Thomas P. Moran, Patrick Chiu, William van Melle, and Gordon Kurtenbach. "Implicit structure for pen-based systems within a freeform interaction paradigm". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '95. Denver, Colorado, USA: ACM Press/Addison-Wesley Publishing Co., 1995, pp. 487–494. isbn: 0201847051. doi: 10.1145/223904.223970. url: https://doi.org/10.1145/223904.223970.

[117] MyScript. *Cross-platform handwriting recognition and interactive ink APIs*. https://developer.myscript.com. Last accessed: 2024-08-01. 2024.

[118] Chris Nash. "The Cognitive Dimensions of Music Notations". In: *Proceedings of the First International Conference on Technologies for Music Notation and Representation*. Ed. by Marc Battier, Jean Bresson, Pierre Couprie, Cécile Davy-Rigaux, Dominique Fober, Yann Geslin, Hugues Genevois, François Picard, and Alice Tacaille. TENOR '15. Paris, France, 2015, pp. 190–202. isbn: 978-2-9552905-0-7.

[119] E. Ng, T. Bell, and A. Cockburn. "Improvements to a pen-based musical input system". In: *Proceedings 1998 Australasian Computer Human Interaction Conference.* OzCHI. 1998, pp. 178–185. doi: 10.1109/OZCHI.1998.732212.

[120] Jakob Nielsen. "Enhancing the Explanatory Power of Usability Heuristics". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '94. Boston, Massachusetts, USA: Association for Computing Machinery, 1994, pp. 152–158. isbn: 0897916506. doi: 10.1145/191666.191729. url: https://doi.org/10.1145/191666.191729.

[121] Bernard A Nijstad, Carsten KW De Dreu, Eric F Rietzschel, and Matthijs Baas. "The dual pathway to creativity model: Creative ideation as a function of flexibility and persistence". In: *European review of social psychology* 21.1 (2010), pp. 34–77.

[122] Donald A Norman. "Natural user interfaces are not natural". In: *interactions* 17.3 (2010), pp. 6–10.

[123] Donald A. Norman and Stephen W. Draper. *User Centered System Design; New Perspectives on Human-Computer Interaction*. USA: L. Erlbaum Associates Inc., 1986. isbn: 0898597811.

[124] O'hara, Kenton and Harper, Richard and Mentis, Helena and Sellen, Abigail and Taylor, Alex. "On the naturalness of touchless: Putting the "interaction" back into NUI". In: *ACM Trans. Comput.-Hum. Interact.* 20.1 (Apr. 2013). issn: 1073-0516. doi: 10.1145/2442106.2442111. url: https://doi.org/10.1145/2442106.2442111.

[125] John V Pavlik. *Journalism and the Metaverse*. Anthem Press, 2024.

[126] Gary Perelman, Marcos Serrano, Christophe Bortolaso, Celia Picard, Mustapha Derras, and Emmanuel Dubois. "Combining Tablets with Smartphones for Data Analytics". In: *Human-Computer Interaction – INTERACT 2019*. Ed. by David Lamas, Fernando Loizides, Lennart Nacke, Helen Petrie, Marco Winckler, and Panayiotis Zaphiris. Cham: Springer International Publishing, 2019, pp. 439–460. isbn: 978-3-030-29390-1. doi: 10.1007/978-3-030-29390-1_24.

[127] Ken Pfeuffer, Jason Alexander, Ming Ki Chong, Yanxia Zhang, and Hans Gellersen. "Gaze-Shifting: Direct-Indirect Input with Pen and Touch Modulated by Gaze". In: *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology*. UIST '15. Charlotte, NC, USA: Association for Computing Machinery, 2015, pp. 373–383. isbn: 9781450337793. doi: 10.1145/

2807442.2807460. url: https://doi.org/10.1145/2807442.2807460.

[128] Ken Pfeuffer, Jason Alexander, and Hans Gellersen. "Partially-Indirect Bimanual Input with Gaze, Pen, and Touch for Pan, Zoom, and Ink Interaction". In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. CHI '16. San Jose, California, USA: Association for Computing Machinery, 2016, pp. 2845–2856. isbn: 9781450333627. doi: 10.1145/2858036.2858201. url: https://doi.org/10.1145/2858036.2858201.

[129] Ken Pfeuffer, Abdullatif Dinc, Jan Obernolte, Radiah Rivu, Yasmeen Abdrabou, Franziska Shelter, Yomna Abdelrahman, and Florian Alt. "Bi-3D: Bi-Manual Pen-and-Touch Interaction for 3D Manipulation on Tablets". In: *The 34th Annual ACM Symposium on User Interface Software and Technology*. UIST '21. Virtual Event, USA: Association for Computing Machinery, 2021, pp. 149–161. isbn: 9781450386357. doi: 10.1145/3472749.3474741. url: https://doi.org/10.1145/3472749.3474741.

[130] Ken Pfeuffer, Ken Hinckley, Michel Pahud, and Bill Buxton. "Thumb + Pen Interaction on Tablets". In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI '17. Denver, Colorado, USA: Association for Computing Machinery, 2017, pp. 3254–3266. isbn: 9781450346559. doi: 10.1145/3025453.3025567. url: https://doi.org/10.1145/3025453.3025567.

[131] Peter Pirolli and Stuart Card. "The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis". In: (2005), pp. 2–4. url: https://analysis.mitre.org/proceedings/Final_Papers_Files/206_Camera_Ready_Paper.pdf.

[132] Peter Pirolli and Ramana Rao. "Table Lens as a Tool for Making Sense of Data". In: *Proceedings of the Workshop on Advanced Visual Interfaces*. AVI '96. Gubbio, Italy: Association for Computing Machinery, 1996, pp. 67–80. isbn: 0897918347. doi: 10.1145/948449.948460. url: https://doi.org/10.1145/948449.948460.

[133] Thammathip Piumsomboon, Adrian Clark, Mark Billinghurst, and Andy Cockburn. "User-Defined Gestures for Augmented Reality". In: *CHI '13 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '13. Paris, France: Association for Computing Machinery, 2013, pp. 955–960. isbn: 9781450319522.

doi: 10.1145/2468356.2468527. url: https://doi.org/10.1145/2468356.2468527.

[134] Symphony Pro. *Music notation & composition app for iPad*. https://symphonypro.net. Last accessed: 2023-11-27. 2023.

[135] Miller Puckette. "Max at Seventeen". In: *Computer Music Journal* 26.4 (2002), pp. 31–43. issn: 01489267, 15315169.

[136] Vijayshankar Raman and Joseph M. Hellerstein. "Potter's Wheel: An Interactive Data Cleaning System". In: *Proceedings of the 27th International Conference on Very Large Data Bases*. VLDB '01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 381–390. isbn: 1558608044.

[137] Gonzalo Ramos, Andy Cockburn, Ravin Balakrishnan, and Michel Beaudouin-Lafon. "Pointing Lenses: Facilitating Stylus Input through Visual-and Motor-Space Magnification". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '07. San Jose, California, USA: Association for Computing Machinery, 2007, pp. 757–766. isbn: 9781595935939. doi: 10.1145/1240624.1240741. url: https://doi.org/10.1145/1240624.1240741.

[138] Ramana Rao and Stuart K. Card. "The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus + Context Visualization for Tabular Information". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '94. Boston, Massachusetts, USA: Association for Computing Machinery, 1994, pp. 318–322. isbn: 0897916506. doi: 10.1145/191666.191776. url: https://doi.org/10.1145/191666.191776.

[139] Jens Rasmussen. *Information Processing and Human-Machine Interaction: An Approach to Cognitive Engineering*. USA: Elsevier Science Inc., 1986. isbn: 0444009876.

[140] Oxford Reference. *Definition of Creativity*. https://dictionary.cambridge.org/us/dictionary/english/creativity. Last accessed: 2024-10-27. 2024.

[141] Thomas Resch. "note for Max — An extension for Max/MSP for Media Arts & music". In: *Proceedings of the International Conference on New Interfaces for Musical Expression*. Daejeon, Republic of Korea: Graduate School of Culture Technology, KAIST, 2013, pp. 210–212. doi: 10.5281/zenodo.1178644. url: http://www.nime.org/proceedings/2013/nime2013_57.pdf.

[142] Thomas Resch and Stefan Bilbao. "Controlling complex virtuel instruments—A setup with note for Max and prepared piano sound synthesis". In: *Proceedings of the International Conference on New Interfaces for Musical Expression*. Brisbane, Australia: Queensland Conservatorium Griffith University, 2016, pp. 295–299. isbn: 978-1-925455-13-7. doi: 10.5281/zenodo.1176108. url: http://www.nime.org/proceedings/2016/nime2016_paper0058.pdf.

[143] Mel Rhodes. "An Analysis of Creativity". In: *The Phi Delta Kappan* 42.7 (1961), pp. 305–310. issn: 00317217. url: http://www.jstor.org/stable/20342603 (visited on 10/27/2024).

[144] Yann Riche, Nathalie Henry Riche, Ken Hinckley, Sheri Panabaker, Sarah Fuelling, and Sarah Williams. "As We May Ink? Learning from Everyday Analog Pen Use to Improve Digital Ink Experiences". In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI '17. Denver, Colorado, USA: Association for Computing Machinery, 2017, pp. 3241–3253. isbn: 9781450346559. doi: 10.1145/3025453.3025716. url: https://doi.org/10.1145/3025453.3025716.

[145] Hans Roels. "Comparing the main compositional activities in a study of eight composers". In: *Musicae Scientiae* 20.3 (2016), pp. 413–435. doi: 10.1177/1029864915624737. url: https://doi.org/10.1177/1029864915624737.

[146] Hugo Romat, Caroline Appert, and Emmanuel Pietriga. "Expressive Authoring of Node-Link Diagrams with Graphies". In: *IEEE Transactions on Visualization and Computer Graphics (TVCG)* 27.4 (2021), pp. 2329–2340. doi: 10.1109/TVCG.2019.2950932.

[147] Hugo Romat, Christopher Collins, Nathalie Henry Riche, Michel Pahud, Christian Holz, Adam Riddle, Bill Buxton, and Ken Hinckley. "Tilt-Responsive Techniques for Digital Drawing Boards". In: *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. UIST '20. Virtual Event, USA: Association for Computing Machinery, 2020, pp. 500–515. isbn: 9781450375146. doi: 10.1145/3379337.3415861. url: https://doi.org/10.1145/3379337.3415861.

[148] Hugo Romat, Nathalie Henry Riche, Ken Hinckley, Bongshin Lee, Caroline Appert, Emmanuel Pietriga, and Christopher Collins. "ActiveInk: (Th)Inking with Data". In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19. Glasgow, Scotland Uk: Association for Computing Machinery, 2019,

pp. 1–13. isbn: 9781450359702. doi: 10.1145/3290605.3300272. url: https://doi.org/10.1145/3290605.3300272.

[149] Hugo Romat, Nicolai Marquardt, Ken Hinckley, and Nathalie Henry Riche. "Style Blink: Exploring Digital Inking of Structured Information via Handcrafted Styling as a First-Class Object". 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: 10.1145/3491102.3501988. url: https://doi.org/10.1145/3491102.3501988.

[150] Hugo Romat, Emmanuel Pietriga, Nathalie Henry-Riche, Ken Hinckley, and Caroline Appert. "SpaceInk: Making Space for In-Context Annotations". In: *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. UIST '19. New Orleans, LA, USA: Association for Computing Machinery, 2019, pp. 871–882. isbn: 9781450368162. doi: 10.1145/3332165.3347934. url: https://doi.org/10.1145/3332165.3347934.

[151] Emilia Rosselli Del Turco and Peter Dalsgaard. ""I Wouldn't Dare Losing One": How Music Artists Capture and Manage Ideas". In: *Proceedings of the 15th Conference on Creativity and Cognition*. C&C '23. Virtual Event, USA: Association for Computing Machinery, 2023, pp. 88–102. isbn: 9798400701801. doi: 10.1145/3591196.3593338. url: https://doi.org/10.1145/3591196.3593338.

[152] Mark Runco. "Creativity". In: *Annual review of psychology* 55 (Feb. 2004), pp. 657–87. doi: 10.1146/annurev.psych.55.090902.141502.

[153] Mark Runco and Garrett Jaeger. "The Standard Definition of Creativity". In: *Creativity Research Journal - CREATIVITY RES J* 24 (Jan. 2012), pp. 92–96. doi: 10.1080/10400419.2012.650092.

[154] Vít Rusnák, Caroline Appert, Olivier Chapuis, and Emmanuel Pietriga. "Designing Coherent Gesture Sets for Multi-Scale Navigation on Tabletops". In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. Montreal QC, Canada: Association for Computing Machinery, 2018, pp. 1–12. isbn: 9781450356206. doi: 10.1145/3173574.3173716. url: https://doi.org/10.1145/3173574.3173716.

[155] Ramik Sadana and John Stasko. “Expanding Selection for Information Visualization Systems on Tablet Devices”. In: *Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces*. ISS ’16. Niagara Falls, Ontario, Canada: Association for Computing Machinery, 2016, pp. 149–158. isbn: 9781450342483. doi: 10.1145/2992154.2992157. url: https://doi.org/10.1145/2992154.2992157.

[156] Advait Sarkar, Andrew D. Gordon, Simon Peyton Jones, and Neil Toronto. “Calculation View: multiple-representation editing in spreadsheets”. In: *IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE, 2018, pp. 85–93. doi: 10.1109/VLHCC.2018.8506584.

[157] Bill N Schilit, Gene Golovchinsky, and Morgan N Price. “Beyond paper: supporting active reading with free form digital ink annotations”. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. 1998, pp. 249–256.

[158] Julia Schwarz, Jennifer Mankoff, and Scott Hudson. “Monte carlo methods for managing interactive state, action and feedback under uncertainty”. In: *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*. UIST ’11. Santa Barbara, California, USA: Association for Computing Machinery, 2011, pp. 235–244. isbn: 9781450307161. doi: 10.1145/2047196.2047227. url: https://doi.org/10.1145/2047196.2047227.

[159] Abigail J. Sellen and Richard H.R. Harper. *The Myth of the Paperless Office*. Cambridge, MA, USA: MIT Press, 2003.

[160] Lei Shi, Rachel Campbell, Peggy Chi, Maria Cirimele, Mike Cleron, Kirsten Climer, Chelsey Q Fleming, Ashwin Ganti, Philippe Gervais, Pedro Gonnet, Tayeb A Karim, Andrii Maksai, Chris Melancon, Rob Mickle, Claudiu Musat, Palash Nandy, Xiaoyu Iris Qu, David Robishaw, Angad Singh, and Mathangi Venkatesan. “Inkeraction: An Interaction Modality Powered by Ink Recognition and Synthesis”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. CHI ’24. New York, NY, USA: Association for Computing Machinery, 2024. isbn: 9798400703300. doi: 10.1145/3613904.3642498. url: https://doi.org/10.1145/3613904.3642498.

[161] Rishabh Singh and Sumit Gulwani. “Learning Semantic String Transformations from Examples”. In: *Proc. VLDB Endow.* 5.8 (Apr. 2012), pp. 740–751. issn: 2150-8097. doi: 10.14778/2212351.2212356. url: https://doi.org/10.14778/2212351.2212356.

[162] Steven M Smith and Thomas B Ward. “Cognition and the creation of ideas”. In: *The Oxford handbook of thinking and reasoning* (2012), pp. 456–474.

[163] Awalin Sopan, Manuel Freier, Meirav Taieb-Maimon, Catherine Plaisant, Jennifer Golbeck, and Ben Shneiderman. “Exploring Data Distributions: Visual Design and Evaluation”. In: *International Journal of Human–Computer Interaction* 29.2 (2013), pp. 77–95. doi: 10.1080/10447318.2012.687676.

[164] Michael Spenke, Christian Beilken, and Thomas Berlage. “FOCUS: The Interactive Table for Product Comparison and Selection”. In: *Proceedings of the 9th Annual ACM Symposium on User Interface Software and Technology*. UIST ’96. Seattle, Washington, USA: Association for Computing Machinery, 1996, pp. 41–50. isbn: 0897917987. doi: 10.1145/237091.237097. url: https://doi.org/10.1145/237091.237097.

[165] Arjun Srinivasan, Bongshin Lee, Nathalie Henry Riche, Steven M. Drucker, and Ken Hinckley. “InChorus: Designing Consistent Multimodal Interactions for Data Visualization on Tablet Devices”. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI ’20. Honolulu, HI, USA: Association for Computing Machinery, 2020, pp. 1–13. isbn: 9781450367080. doi: 10.1145/3313831.3376782. url: https://doi.org/10.1145/3313831.3376782.

[166] StaffPad. *Music composition app designed for pen and touch*. https://www.staffpad.net. Last accessed: 2023-11-27. 2023.

[167] R. Sternberg. “The Nature of Creativity”. In: *Creativity Research Journal* 18 (Jan. 2006), pp. 87–98. doi: 10.1207/s15326934crj1801_10.

[168] Robert J. Sternberg and Todd Lubart. “The concept of creativity: Prospects and paradigms.” In: 1998. url: https://api.semanticscholar.org/CorpusID:147240429.

[169] Anselm Strauss and Juliet Corbin. *Basics of qualitative research: Techniques and procedures for developing grounded theory, 2nd ed.* Sage Publications, 1998. isbn: 9780803959408.

[170] Hari Subramonyam and Eytan Adar. “SmartCues: A Multitouch Query Approach for Details-on-Demand through Dynamically Computed Overlays”. In: *IEEE Transactions on Visualization and Computer Graphics* PP (Aug. 2018), pp. 1–1. doi: 10.1109/TVCG.2018.2865231.

[171] Hariharan Subramonyam, Colleen Seifert, Priti Shah, and Eytan Adar. “texSketch: Active Diagramming through Pen-and-Ink Annotations”. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI ’20. Honolulu, HI, USA: Association for Computing Machinery, 2020, pp. 1–13. isbn: 9781450367080. doi: 10.1145/3313831.3376155. url: https://doi.org/10.1145/3313831.3376155.

[172] J. Sweller, P. Ayres, and S. Kalyuga. *Cognitive Load Theory*. Explorations in the Learning Sciences, Instructional Systems and Performance Technologies. Springer New York, 2011. isbn: 9781441981257. url: https://books.google.fr/books?id=H4qQNQEACAAJ.

[173] Yuta Takayama, Yuu Ichikawa, Buntarou Shizuki, Ikkaku Kawaguchi, and Shin Takahashi. “A User-Based Mid-Air Hand Gesture Set for Spreadsheets”. In: *Asian CHI Symposium 2021*. Asian CHI Symposium 2021. Yokohama, Japan: Association for Computing Machinery, 2021, pp. 122–128. isbn: 9781450382038. doi: 10.1145/3429360.3468193. url: https://doi.org/10.1145/3429360.3468193.

[174] Poorna Talkad Sukumar, Anqing Liu, and Ronald Metoyer. “Replicating User-Defined Gestures for Text Editing”. In: *Proceedings of the 2018 ACM International Conference on Interactive Surfaces and Spaces*. ISS ’18. Tokyo, Japan: Association for Computing Machinery, 2018, pp. 97–106. isbn: 9781450356947. doi: 10.1145/3279778.3279793. url: https://doi.org/10.1145/3279778.3279793.

[175] OSMD Team. *OpenSheetMusicDisplay*. https://opensheetmusicdisplay.org. Last accessed: 2024-08-01. 2024.

[176] Meinald Thielsch and Isabel Perabo. “Use and Evaluation of Presentation Software”. In: *Technical Communication* (May 2012).

[177] Martin Tricaud and Michel Beaudouin-Lafon. “Revisiting creative behaviour as an epistemic process: lessons from 12 computational artists and designers”. In: *Proceedings of the 35th Australian Computer-Human Interaction Conference*. OzCHI ’23. Wellington, New Zealand: Association for Computing Machinery, 2024, pp. 175–190. isbn: 9798400717079. doi: 10.1145/3638380.3638395. url: https://doi.org/10.1145/3638380.3638395.

[178] Theophanis Tsandilas. “Fallacies of Agreement: A Critical Review of Consensus Assessment Methods for Gesture Elicitation”. In: *ACM Trans. Comput.-Hum. Interact.* 25.3 (June 2018). issn: 1073-0516. doi: 10.1145/3182168. url: https://doi.org/10.1145/3182168.

[179] Theophanis Tsandilas, Catherine Letondal, and Wendy E. Mackay. “Musink: Composing Music through Augmented Drawing”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’09. Boston, MA, USA: Association for Computing Machinery, 2009, pp. 819–828. isbn: 9781605582467. doi: 10.1145/1518701.1518827. url: https://doi.org/10.1145/1518701.1518827.

[180] Daniel Vogel and Ravi Balakrishnan. “Direct Pen Interaction With a Conventional Graphical User Interface”. In: *Human–Computer Interaction* 25.4 (2010), pp. 324–388. doi: 10.1080/07370024.2010.499839.

[181] Daniel Vogel and Patrick Baudisch. “Shift: a technique for operating pen-based interfaces using touch”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’07. San Jose, California, USA: Association for Computing Machinery, 2007, pp. 657–666. isbn: 9781595935939. doi: 10.1145/1240624.1240727. url: https://doi.org/10.1145/1240624.1240727.

[182] Daniel Vogel, Matthew Cudmore, Géry Casiez, Ravin Balakrishnan, and Liam Keliher. “Hand occlusion with tablet-sized direct pen input”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’09. Boston, MA, USA: Association for Computing Machinery, 2009, pp. 557–566. isbn: 9781605582467. doi: 10.1145/1518701.1518787. url: https://doi.org/10.1145/1518701.1518787.

[183] Jagoda Walny, Bongshin Lee, Paul Johns, Nathalie Henry Riche, and Sheelagh Carpendale. “Understanding Pen and Touch Interaction for Data Exploration on Interactive Whiteboards”. In: *IEEE Transactions on Visualization and Computer Graphics* 18.12 (2012), pp. 2779–2788. doi: 10.1109/TVCG.2012.275.

[184] Peter R Webster. “Creativity and music education”. In: ed. by T. Sullivan and L. Willingham. Canadian Music Educators’ Association, 2002. Chap. Creative thinking in music: Advancing a model, pp. 16–33.

[185] Kirsten N. Whitley and Alan F. Blackwell. "Visual programming: the outlook from academia and industry". In: *Workshop on Empirical Studies of Programmers*. 1997. url: https://api.semanticscholar.org/CorpusID:18983760.

[186] Daniel Wigdor and Dennis Wixon. *Brave NUI world: designing natural user interfaces for touch and gesture*. Elsevier, 2011.

[187] Wesley Willett, Jeffrey Heer, and Maneesh Agrawala. "Scented Widgets: Improving Navigation Cues with Embedded Visualizations". In: *IEEE Transactions on Visualization and Computer Graphics* 13.6 (2007), pp. 1129–1136. doi: 10.1109/TVCG.2007.70589.

[188] Andrew D Wilson, Shahram Izadi, Otmar Hilliges, Armando Garcia-Mendoza, and David Kirk. "Bringing physics to the surface". In: *Proceedings of the 21st annual ACM symposium on User interface software and technology*. 2008, pp. 67–76.

[189] Megan A. Winget. "Heroic Frogs Save the Bow: Performing Musician's Annotation and Interaction Behavior with Written Music". In: *Proceedings of the 7th International Conference on Music Information Retrieval*. Victoria, Canada: ISMIR, Oct. 2006, pp. 73–78.

[190] Jacob O. Wobbrock, Leah Findlater, Darren Gergle, and James J. Higgins. "The Aligned Rank Transform for Nonparametric Factorial Analyses Using Only Anova Procedures". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '11. Vancouver, BC, Canada: Association for Computing Machinery, 2011, pp. 143–146. isbn: 9781450302289. doi: 10.1145/1978942.1978963. url: https://doi.org/10.1145/1978942.1978963.

[191] Jacob O. Wobbrock, Meredith Ringel Morris, and Andrew D. Wilson. "User-defined Gestures for Surface Computing". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI'09. Boston, MA, USA: ACM, 2009, pp. 1083–1092. isbn: 978-1-60558-246-7. doi: 10.1145/1518701.1518866. url: http://doi.acm.org/10.1145/1518701.1518866.

[192] Richard W. Woodman, John E. Sawyer, and Ricky W. Griffin. "Toward a Theory of Organizational Creativity". In: *The Academy of Management Review* 18.2 (1993), pp. 293–321. issn: 03637425. url: http://www.jstor.org/stable/258761 (visited on 11/13/2024).

[193] Mike Wu and Ravin Balakrishnan. "Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays". In: *Proceedings of the 16th annual ACM symposium on User interface software and technology*. 2003, pp. 193–202.

[194] Haijun Xia, Nathalie Henry Riche, Fanny Chevalier, Bruno De Araujo, and Daniel Wigdor. “DataInk: Direct and Creative Data-Oriented Drawing”. In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. Montreal QC, Canada: Association for Computing Machinery, 2018, pp. 1–13. isbn: 9781450356206. doi: 10.1145/3173574.3173797. url: https://doi.org/10.1145/3173574.3173797.

[195] Haijun Xia, Ken Hinckley, Michel Pahud, Xiao Tu, and Bill Buxton. “WritLarge: Ink Unleashed by Unified Scope, Action, & Zoom”. In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI '17. Denver, Colorado, USA: Association for Computing Machinery, 2017, pp. 3227–3240. isbn: 9781450346559. doi: 10.1145/3025453.3025664. url: https://doi.org/10.1145/3025453.3025664.

[196] Kai Xiong, Siwei Fu, Guoming Ding, Zhongsu Luo, Rong Yu, Wei Chen, Hujun Bao, and Yingcai Wu. “Visualizing the Scripts of Data Wrangling With Somnus”. In: *IEEE Transactions on Visualization and Computer Graphics* 29.6 (2023), pp. 2950–2964. doi: 10.1109/TVCG.2022.3144975.

[197] Ka-Ping Yee. “Two-Handed Interaction on a Tablet Display”. In: *CHI '04 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '04. Vienna, Austria: Association for Computing Machinery, 2004, pp. 1493–1496. isbn: 1581137036. doi: 10.1145/985921.986098. url: https://doi.org/10.1145/985921.986098.

[198] Dongwook Yoon, Nicholas Chen, and François Guimbretière. “TextTearing: opening white space for digital ink annotation”. In: *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*. UIST '13. St. Andrews, Scotland, United Kingdom: Association for Computing Machinery, 2013, pp. 107–112. isbn: 9781450322683. doi: 10.1145/2501988.2502036. url: https://doi.org/10.1145/2501988.2502036.

[199] Dongwook Yoon, Nicholas Chen, François Guimbretière, and Abigail Sellen. “RichReview: Blending Ink, Speech, and Gesture to Support Collaborative Document Review”. In: *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*. UIST '14. Honolulu, Hawaii, USA: Association for Computing Machinery, 2014, pp. 481–490. isbn: 9781450330695. doi: 10.1145/2642918.2647390. url: https://doi.org/10.1145/2642918.2647390.

[200] Dimitrios Zbainos and Todd Lubart. *Creativity*. Apr. 2020. doi: 10.1093/acrefore/9780190264093.013.895. url: https://oxfordre.com/education/view/10.1093/acrefore/9780190264093.001.0001/acrefore-9780190264093-e-895.

[201] Robert Zeleznik, Andrew Bragdon, Ferdi Adeputra, and Hsu-Sheng Ko. "Hands-on Math: A Page-Based Multi-Touch and Pen Desktop for Technical Work and Problem Solving". In: *Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology*. UIST '10. New York, New York, USA: Association for Computing Machinery, 2010, pp. 17–26. isbn: 9781450302715. doi: 10.1145/1866029.1866035. url: https://doi.org/10.1145/1866029.1866035.

[202] Emanuel Zgraggen, Robert Zeleznik, and Steven M. Drucker. "PanoramicData: Data Analysis through Pen and Touch". In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pp. 2112–2121. doi: 10.1109/TVCG.2014.2346293.

[203] Emanuel Zgraggen, Robert Zeleznik, and Philipp Eichmann. "Tableur: Handwritten Spreadsheets". In: *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. CHI EA '16. San Jose, California, USA: Association for Computing Machinery, 2016, pp. 2362–2368. isbn: 9781450340823. doi: 10.1145/2851581.2892326. url: https://doi.org/10.1145/2851581.2892326.

[204] Yang Zhang, Michel Pahud, Christian Holz, Haijun Xia, Gierad Laput, Michael McGuffin, Xiao Tu, Andrew Mittereder, Fei Su, William Buxton, and Ken Hinckley. "Sensing Posture-Aware Pen+Touch Interaction on Tablets". In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19. Glasgow, Scotland Uk: Association for Computing Machinery, 2019, pp. 1–14. isbn: 9781450359702. doi: 10.1145/3290605.3300285. url: https://doi.org/10.1145/3290605.3300285.

Chapter A

Appendices

A.1 Chapter 3: Elicitation Study

A.1.1 List of referents

Table A.1 lists questions for the 28 referents considered in the elicitation study. Some questions were actually a bit more detailed to give context (*e.g.*, the name of the columns to merge for GM_1). Study material is available as supplemental material.

A.1.2 Definition of a Sign

We define a *sign* as a series of events that is described along the following dimensions:

- The input modality, which can be Pen tip, Pen eraser, Single Touch, Multi-touch or Pen + Touch.
- The start and end locations of input, which can be a Column header, a Row header, a Cell, somewhere Inside-a-Cell, the Select-All button, a Column separator, the Background. We use Inside-a-Cell when the location within the cell itself carries information (*e.g.*, the participant draws a line between two specific characters of the value string).
- The input event type. We use four types of discrete events: Tap, Double Tap, Dwell and Flick. For continuous events, if the trace's trajectory does not bear meaningful information, we classify it as Drag. For other continuous events, we use the following five categories: Vertical Line, Horizontal Line, Diagonal Line, Enclose or ZigZag. A few traces do not fall in any of those categories and rather correspond to custom-shape gestures that we categorize into one of the following shapes: Circle, Arrow, Equal sign, Parallel sign, Less-than sign, V, Loops.

An event is defined as a combination of these dimensions, and a *sign* can be either a single event or a combination of atomic events. Our definition of a sign is quite specific not only regarding the description of an event but also regarding the transition between consecutive events. In particular, when a sign involves a couple of events that have the same modality, we make a distinction between the case where the input device remains in contact with the screen during the transition, and the case where it is lifted up between the two events. For example, a Dwell immediately followed by a Drag without lifting the pen up is different from a Dwell + Drag sequence where the user lifts the pen up after the Dwell. For the coarser *modality*-based classification, a participant's proposal is simply described as the combination of its events' modalities.

A.2 Chapter 3: Implementation Details

A.2.1 Prototype Implementation

The Web-based prototype depicted in Figure 3.15 and used for the semi-structured qualitative study implements all interaction techniques from Section 3.4. It is developed entirely in JavaScript and D3 [21], runs on the client side. Spreadsheet elements and interface widgets are all rendered in SVG. User pen and touch input events are handled with the W3C Pointer API [26].

The prototype is made available as supplemental material, and has been tested extensively with the Chromium Web browser on a Windows 10 PC connected to a Wacom Cintiq Pro. It also runs for instance on a Microsoft Surface Studio 2+, although some interactions that involve two simultaneous contact points are not supported so far because of input event API compatibility issues (the level of support for the W3C pointer API varies significantly across Web browsers and operating systems).

A.2.2 Generalizing Subcell Selections

Algorithm 1 below details how generalization works for subcell selections that include the cell's first character. Informally, priority is given to special characters such as dash, comma, *etc.*, falling back to different alphanumeric transitions (including juxtapositions of uppercase and lower case letters in either order) if no such character could be found. Other cases work similarly but are not detailed for the sake of conciseness: selections that include the last character use a mirror of the algorithm below; selections that include neither the first nor the last character use a combination of both algorithms; selections of the latter category consisting of a single character are generalized based on the transition from the previous character rather than the next one, consistent with the reading direction.

Algorithm 1: Generalization of a subcell selection that includes the first character of the source string. String indices start at 1. $s[i]$ returns the character at index i in string s. $s[i : j]$ returns all characters from string s between indices i and j included.

```
Def: 𝒟    // set of all delimiters, including special characters, arithmetic operators and currency symbols
Def: ℒ_l                          // set of all lowercase letters
Def: ℒ_u                          // set of all uppercase letters
Def: 𝒩                            // set of all digits
Def: enum U2L, L2U, A2N, N2A  // transitions from/to upper & lower case, from/to letter & number

Data: s                           // source cell (string)
Data: 𝒯                           // set of target cells (strings)
Data: i_s            // index of last char ∈ source subcell selection

ToI ← <pos: 0, type: None>   // transition of interest: <position, type>
if s[i_s] ∈ 𝒟 or s[i_s + 1] ∈ 𝒟 then
    ToI ← <pos: i_s, type: s[i_s]>              // special character
else
    if s[i_s] ∈ ℒ_u ∪ ℒ_l then
        if s[i_s + 1] ∈ ℒ_u ∪ ℒ_l then
            if s[i_s] ∈ ℒ_u and s[i_s + 1] ∈ ℒ_l then
                ToI.type ← U2L          // switch upper → lower case
                ToI.pos ← count(U2L, s[1 : i_s − 1])
            else if s[i_s] ∈ ℒ_l and s[i_s + 1] ∈ ℒ_u then
                ToI.type ← L2U          // switch lower → upper case
                ToI.pos ← count(L2U, s[1 : i_s − 1])
        else
            ToI.type ← A2N                  // switch letter → number
            ToI.pos ← count(A2N, s[1 : i_s − 1])
    else
        if s[i_s + 1] ∈ ℒ_u ∪ ℒ_l then
            ToI.type ← N2A                  // switch number → letter
            ToI.pos ← count(N2A, s[1 : i_s − 1])
if ToI.type ≠ None then
    for t ∈ 𝒯 do
        i_t ← indexOf(ToI, t)   // get index of n^th occurence of ToI.type where n = ToI.pos
        if i_t > 0 then
            select t[1 : i_t]   // select up to index of n^th ToI occurence in t
else
    for t ∈ 𝒯 do
        select t[1:ToI.pos]   // no delimiter identified, select based on original selection length
```

Scope	Action Type	Question
Value-level	Selection	(VS_1) How would you select the first character of a string in a cell? (VS_2) How would you select the comma (and only the comma character) in a cell? (VS_3) How would you select the last character of a string in a cell? (VS_4) How would you select the left part of a string in a cell? (VS_5) How would you select the sequence of characters `", NY"` (and only that sequence) in a cell? (VS_6) How would you select the right part of a string in a cell? (VS_7) How would you generalize a sub-cell selection to its parent column?
	Manipulation	(VM_1) How would you move a selection within a cell? (VM_2) How would you delete part of the content of a cell? (VM_3) How would split a column into two columns?
Grid-level	Selection	(GS_1) How would you select a cell? (GS_2) How would you select a range of cells? (GS_3) How would you select a column? (GS_4) How would you select a range of columns? (GS_5) How would you select a set of columns? (GS_6) How would you select a row? (GS_7) How would you select a range of rows? (GS_8) How would you select a set of rows? (GS_9) How would you select the set of cells that have the same value in a column? (GS_{10}) How would you select the set of rows that have the same value for a specific cell?
	Manipulation	(GM_1) How would you merge two columns into one? (GM_2) How would you move a column? (GM_3) How would you move a row? (GM_4) How would you clear a cell? (GM_5) How would you delete a column? (GM_6) How would you delete a row? (GM_7) How would you sort a column? (GM_8) How would you fill up a column following the pattern of selected values?

Table A.1: Referents considered in the elicitation study.