An evaluation of the direct manipulation techniques used in Lyra

Christina Müller
Human-IST Institute
University of Fribourg, Switzerland
christina.mueller@students.unibe.ch

ABSTRACT

Lyra is a declarative and interactive environment for designing custom visualizations [5]. Satyanarayan and Heer [5] showed that Lyra is expressive and accessible. Accessibility can be interpreted as direct manipulation property or directness of a tool. In this paper, I evaluate which direct manipulation techniques are observable in Lyra. I find that Lyra has lots of properties which contribute to its directness. Data transformation is one part of Lyra which does not use direct manipulation techniques and therefore lack directness. It is arguable how many and which direct manipulation techniques are best and further research should be done to empirically prove what generates directness.

KEYWORDS

Lyra, direct manipulation, feeling of directness

1 INTRODUCTION

Lyra is a declarative, interactive, direct manipulation environment for designing static custom visualizations [5].
In a direct manipulation system, user interaction takes place graphically and matches the way a user thinks. This reduces the cognitive load for the user. [2]
In section 3, I define directness and how it can be achieved based on the work of [2]. In section 4, I evaluate how these theoretical concepts are applied in Lyra. In section 5, I make a reflection on the made evaluation.
The contribution of this paper is to better understand why Lyra produces a feeling of directness in the user and what could be changed to increase directness. Furthermore, the transfer from theory to practice may be used for the design and evaluation of other visualization design environments.

2 RELATED WORK

Satyanarayan and Heer [5] showed that Lyra is expressive and accessible. They implicitly defined *accessible* as offering direct manipulation techniques. Usually there is a trade-off between these two properties. Programming-based tools for example are highly flexible and expressive but they have the disadvantage of being time-consuming, error-prone and only understandable by experts. [5]
[5] created data visualizations with Lyra to show that the expressiveness of Lyra is comparable to programming-based frameworks. They evaluated the accessibility of Lyra by a first-use study including graphic designers with different programming knowledge, but did not show how Lyra achieves the property of being accessible.
Hutchins, Hollan and Norman [2] described from a theoretical point of view how a system can generate feeling of directness in the user.
Kandel, Paepcke, Hellerstein and Heer [3] showed how to use direct manipulation techniques for data transformations.
Saket, Srinivasan, Ragan and Endert [4] studied the accuracy of the user when manipulating graphical encodings.
My work is based on these last three approaches.

3 DIRECTNESS IN THEORY

Definition of directness

Direct manipulation or synonomously *(producing a feeling of) directness* is a property of a tool, which means that user interaction takes place graphically and matches the way a user thinks and therefore reduces the cognitive load for the user. Features of a tool which result in directness are called *direct manipulation techniques*. [2]
[6] suggests that a direct manipulation interface should be suitable for novices as well as for experts.
Directness of an interface depends on the task of the user [2]. I consider directness of Lyra in relation to the task of creating data visualizations.

Producing feeling of directness

Distance and (direct) engagement are two design aspects which influence directness of an interface. Also adaptation by and learning of a user produce a feeling of directness. Because the last two aspects are not under the control of a tool designer I do not consider them. [2]

The first design aspect – *distance* – refers to the gap between the user's intention and the facilities and language of the system. It has two directions: [2]

- *gulf of execution*: refers to the gap from the user to the system. It describes how easily a user can realize its intent.

- *gulf of evaluation*: refers to the gap from the system to the user. It describes how easily the user can perceive and interpret the output of the system in relation to its goal.

Small gulf of execution and evaluation means small distance and therefore increases directness.
There are two types of distance: semantic and articulatory distance of a tool: [2]

- *semantic distance*: refers to the transformation of the user's language into the system's language and vica versa. High-level language, the display of necessary things only and consistency contribute to a small semantic distance and therefore reduce the gulf of execution and evaluation. This results in high directness. High-level language means that the user does not have to decompose the task into low-level operations and that the tool's language matches the task domain.
- *articulatory distance*: refers to the similarity between the meaning and the physical form of the tool's language. An example of small articulatory distance is the increment of a variable by moving a vertical slider up. A small articulatory distance reduces the gulf of execution and evaluation and therefore increases directness.

The second design aspect – *engagement* – means that objects are represented in a way so that the representations feel like the objects themselves [2]. According to [2] and [6] the following design choices result in high engagement:

- The user can directly act on objects.
- Input and output refer to each other: Input will be transformed to output and this output can be a component of a new input.
- The user does not perceive the interface.
- Actions are reversible.
- The user's action has direct visual impact with no delays between the action and the displayed results.

4 EVALUATION OF DIRECTNESS IN LYRA

I evaluate if and how the design aspects of distance and engagement described in the section before are observable in Lyra. The evaluation is summarized in Table 1.
In some cases, I illustrate my points with screenshots of features in Lyra. Because some readers – mainly Lyra experts – may understand the arguments without visual clarification, the figures are placed in the appendix.

Evaluation of semantic distance

Several features in Lyra decrease the semantic distance. First, Lyra uses a high-level language:

- In most cases, Lyra is intuitively understandable: It uses natural-language. There are icons with tooltips (Figure 8) and small descriptions such as for the scale delete button (Figure 6).
- The user has not to decompose the task in low-level operations. For example the interaction with the canvas does not require any knowledge of a programming language, it includes among others drag-and-drop.

Second, only necessary things are displayed: In the left and right panel only those pipelines, layers and axes or marks the user is currently interacting with are shown in detail. Third, Lyra tries to be consistent, for example it always uses + for adding and − for deleting. Fourth, features as marks, drop zones, connectors and handles (Figure 1) make actions easier, for example via drag-and-drop. Fifth, there is automation: Lyra automatically creates scales if the user appends data to marks (Figure 2) [5]. These first five points contribute to a small gulf of execution.
The semantic distance is further decreased by the visualization of the output in the canvas and the coloring of the data in the DATA PIPELINE: The origin data set is colored green, data achieved by transformation is yellow (Figure 4). These last two points both contribute to a small gulf of evaluation.
Some parts of Lyra's architecture increase the semantic distance. First, for data transformations, the user has to give input in low-level language. For example, the conditional operator `? :` has to be used to express an if else data transformation (Figure 4). Or for filtering based on more than one criterion the boolean operators `&&` and `||` have to be used (Figure 5). Second, not every intent of the user can be executed, Lyra offers only RGB color space and there are no arc marks or radial axes available [5]. Third, there are inconsistencies which may make the process of translating the intent into actions more difficult. Consider the concept of guides. This concept would be simple: axes for graphical domains and legends for scales like color, shape and size [5]. But we have the following inconsistency: axes can be inspected via the right panel whereas it is not obvious how to add and manipulate legends. Fourth, the study of [5] showed that Lyra does not infer a user's intent when he misses a drop zone by a few pixels. Fifth, there are a lot of terms and objects in Lyra: marks, visual layouts, axes, scales and its type, domain and range which meanings may not be intuitively clear. This can confuse the user and therefore increase the semantic distance.

Evaluation of articulatory distance

Articulatory distance is decreased by several design choices: The layers which belong to different coordinates and z-indexes in the canvas are displayed as a nested list in the VISUALIZATION panel. This nested structure represents the architecture of the layers. In addition, the user can select the color of the axes, areas etc. via a color inspector (Figure 10).

Table 1: Direct manipulation techniques in Lyra.

	directness increased by	directness decreased by
Distance	small semantic distance	high semantic distance
	(1) high-level language: tooltips, explanations, interaction in canvas (2) display of necessary things only: layers, axes etc. can be hidden (3) consistency: − for deleting, + for adding (4) marks, drop zones, connectors and handles (5) automation: scale generation (6) output visible in canvas (7) colored data	(1) low-level language for data transformations (2) available options restricted (3) inconsistencies: axes vs legends (4) missed drop zones (5) lots of terms
	small articulatory distance	high articulatory distance
	(1) layers as nested list in VISUALIZATION panel (2) color selection via color inspector	(1) font suggestions do not represent font (2) drop zones on top and on the right of a mark only (3) − for deleting and not for hiding
Engagement		
	(1) user acts on object: operations and manipulating objects in canvas, process of deletion (2) input = output: manipulation in canvas (3) interactivity hides interface (4) in general reversible and rapid operations with direct visible output in canvas	(1) user has to describe action in case of the deleting a scale (2) interface visible through panels, display of raw data and low-level language for transformation (3) sometimes slow operations and no visible impact (e.g. no preview of data transformations)

There are some features which contribute to a high articulatory distance. First, the suggested fonts (e.g. for the title of an axis) are not directly written in the font style itself (Figure 7). Second, drop zones only appear on the top and on the right of a mark. Some people may expect the horizontal drop zone at the bottom and the vertical one on the left of the mark. Third, Lyra uses the − as a button for deleting elements and the + for adding (Figure 9). This contributes to consistency, but the problem may be that some people associate the − with hiding and may click on that button instead on the element itself to hide and show it.

Evaluation of engagement

Several features of Lyra increase its engagement: the user can directly act on the elements of the visualization via the canvas. There are marks, drop zones, connectors and handles to support these actions (Figure 1). Also most of the elements (e.g. layers) can be deleted directly by clicking on the − next to the element itself. The scale is one exception. The menu of a scale has to be opened first and then the delete button appears. This process does not relate to the real life where you can put your hand on an object, grab it and place it in the bin. So for deletion of scales, engagement is low.
For manipulations made in the canvas, input is equal to output. This contributes to high engagement. Also when properties of elements such as marks are changed in the side panels, the output (= change of a property of the mark) will directly appear on the input object (= mark).
Lyra is a highly interactive tool, this involvement of the user makes the interface less visible and also contributes to high engagement. On the other hand, the division of the tool in three panels, the display of raw data and the low-level language for transformation may have a contrary effect.
Due to the undo button, actions are reversible in Lyra. Furthermore, Lyra is rapid and user action has direct visible impact in the canvas. These factors contribute to high engagement. Only in a few cases, the actions are slow or their output is not directly visible in the canvas. Two examples:

- The switching of drop zones when moving the mouse from one mark to another is slow and also the adaption of the width of an axis via the VISUALIZATION panel is slow compared to the other processes in Lyra.
- Some data transformations are only applied on the data table and not directly on the data in the canvas. For example if the user adds a formula, a new data row gets appended but data in the visualization does not get adapted whereas for other transformations it does.

5 REFLECTION ON THE MADE EVALUATION

Lyra includes many direct manipulation techniques. This may explain why Lyra is described as "natural" and "intuitive" by the users in the study of [5].

Data transformation in Lyra

Compared to the other drawbacks observed in section 4, data transformation is a big part of Lyra where direct manipulation techniques are missing. Since data transformation is the fundamental and first step of every data visualization and can be very time-consuming [3], it is worth to consider how Lyra could increase directness in this part.

Unlike Lyra, the interactive and declarative tool Wrangler offers direct manipulation techniques. *Wrangler* provides eight data transformations: map, lookups and joins, reshape, positional, sorting, aggregation, key generation and schema transforms. The interface of Wrangler includes a header and two panels. The left panel contains the history of transformations and the suggested transformations. The right panel displays the data in an interactive table. The header includes different actions such as delete, split and fill. [1, 3]
I mention some features of Wrangler to show how Lyra could increase directness when it comes to data transformation.

- Decrease semantic distance: To reduce the gulf of execution, Wrangler suggests data transformation based on the user input, the type or semantic role of the data and empirically-derived heuristics. User input can be given by choosing an action from the menu or by modifying the data and suggested transformations. Based on the user input Wrangler refines the set of suggestions. Natural language descriptions and a visual preview of the suggested transformation are provided. This high-level language features reduce the gulf of evaluation. [1, 3]
- Increase engagement: Users can directly interact with the data table. For example, rows and columns can be selected. This increases the feeling of engagement. [1]

[3] showed that Wrangler significantly reduces the time for data transformations compared to other data manipulation tools (Microsoft Excel). The participants of their study had never used Wrangler before and the median score of prior experience with Excel on a 10-point scale was 5.

How many and which direct manipulation techniques

It is not the goal of this paper to quantitatively judge how many features that contribute to the feeling of directness are implemented in Lyra. How many and which manipulation techniques are appropriate is a complex question and the answer depends on the context [2]. Related to this question I summarize two important design suggestions:
First, it is not always best to apply as many direct manipulation techniques as possible. According to [2] direct manipulation techniques may be unfavorable in some situations because they do not offer options to facilitate repetitive tasks and they decrease the generality, flexibility and accuracy of a tool.
Second, designers of data visualization tools also carefully have to choose which direct manipulation technique to apply when. [4] showed that when a user has to change the intensity of a graphical encoding (position, distance from a circle to a rectangle, length of rectangles, area of squares and circles, curvature, shading and angle) in proportion to an initial value, changing the position of a slider was executed with significantly more accuracy than all other changes of graphical encodings. Lyra implements these results: First, for graphical encoding in the side panels Lyra mainly uses the positioning: the width of axes or marks, the opacity of fillings and the padding for the ordinal scale type all can be changed via a positioning slider. Second, in addition to the direct manipulation of the objects in the canvas and the positioning slider, Lyra has the option to edit properties via numerical value input in the VISUALIZATION panel. This makes data visualization more expressive and precise (Figure 3).

6 OPEN RESEARCH QUESTION

This paper is an application of the theoretical concepts of [2] in the tool Lyra. In section 4, I evaluate what features of Lyra contribute to small distance and high engagement and therefore explain the high feeling of directness in the user produced by Lyra. It is arguable whether it can be empirically proven that these features result in high directness. Probably, it is difficult to empirically measure the influence of one single property of a tool on the feeling of directness. But it could be researched what influence groups of features (e.g. a group of features which claims to reduce semantic distance or articulatory distance or a group of features which claims to increase engagement) have on the feeling of directness. Also it is arguable whether there are dependencies between the semantic distance, articulatory distance and engagement related to the increment of directness.

7 CONCLUSION AND FUTURE WORK

Lyra uses different direct manipulation techniques. This may explain why Lyra is judged as accessible in the work of [5]. There are options to improve direct manipulation and therefore the feeling of directness in Lyra for example when it comes to data transformations. The tool Wrangler gives examples of how Lyra could apply direct manipulation techniques for data transformation.
But it is not always best to apply as many direct manipulation techniques as possible. Because too many and the wrong direct manipulation techniques are disadvantageous for some tasks and can decrease the expressiveness of a tool. It is arguable how many and which direct manipulation techniques are best in which situations and further research should be done to empirically prove what generates directness.
The evaluation of Lyra's directness is mainly based on the paper of [2], written in 1985 and on the author's view. Further evaluation could be based on a broader and – if there will exist any in the future – newer theoretical fundament and be done in collaboration with multiple novices and experts.

A APPENDICES

The appendix includes screenshots of Lyra (http://idl.cs.washington.edu/projects/lyra/app/, last retrieved on 20.04.2018) referred to in section 4 and 5.

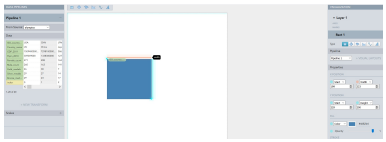

Figure 1: Marks, drop zones, connectors and handles. A *mark* is a shape, e.g. a rectangle, to which data can be added. Marks are available in the menu at the top of the canvas and can be placed in the canvas via drag-and-drop. When a row of the data table in the DATA PIPELINE is dragged to a mark, *drop zones* appear. These are areas where data can be placed. Marks can be relatively placed next to each other with the help of diamond shaped points – so called *connectors*. *Handles* appear if the user clicks on a mark, the user can move, rotate and resize a mark via these handles. [5]

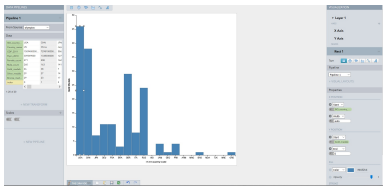

Figure 2: Automatic bar plot and scale generation. In this example, ISO country codes are placed on the top and the number of gold medals for each country on the side of a rectangle. Lyra then automatically creates a bar plot and the related scales. [5]

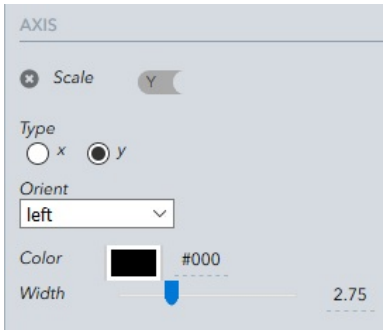

Figure 3: Width manipulation. Manipulating the width of an axis in Lyra's VISUALIZATION panel via positioning of a slider or via numerical value input.

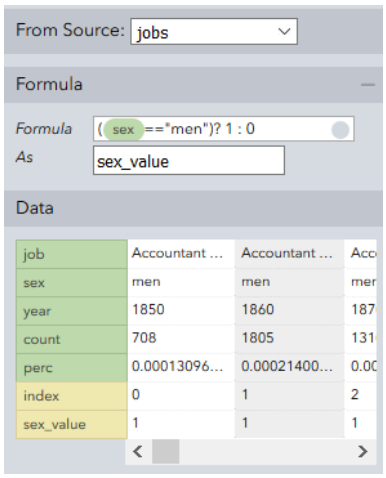

Figure 4: Conditional operator. Doing data transformation via conditional operator in Lyra. The output of the transformation is appended as a new row to the data table and colored yellow.

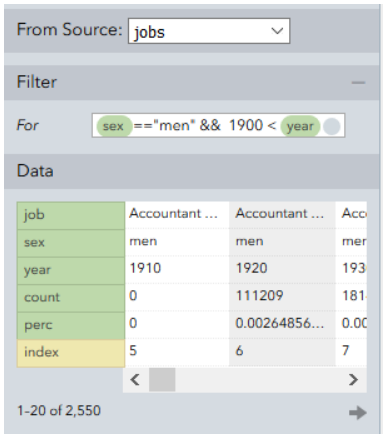

Figure 5: Boolean operator. Doing data transformation via boolean operator in Lyra.

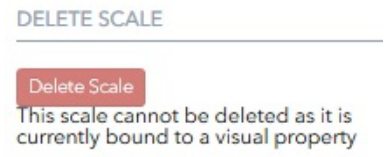

Figure 6: Small description. Description of the Delete Scale button in Lyra.

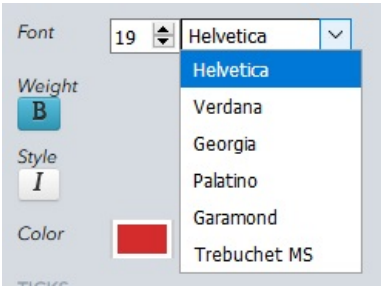

Figure 7: Font style selection. Font suggestions in Lyra are not written in the font style itself.

Figure 8: Tooltips. Tooltips for explanations in Lyra.

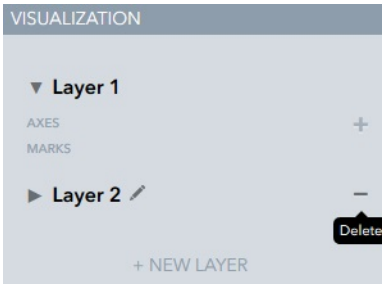

Figure 9: Hiding, showing, deleting and adding objects. In Lyra, a detailed view of the objects can be hidden and shown via mouse-click on the object itself. The − is for deleting, the + for adding objects.

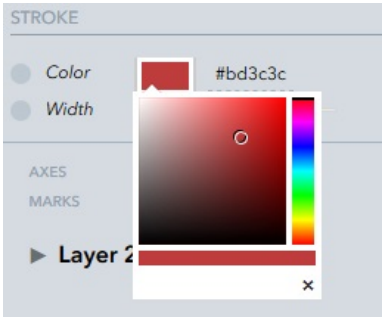

Figure 10: Color selection. In Lyra, the color of elements can be chosen via color inspector.

ACKNOWLEDGEMENTS

This paper is a student work, written for Seminar SS2018 "Data Visualization", under the supervision of Prof. Denis Lalanne and Florian Evequoz. Human-IST Institute, University of Fribourg, Switzerland, 2018.
I thank the supervisors for their encouragement and I also thank Kathrin Blümli for checking spelling and grammar and Luca Liechti for reviewing this paper.

REFERENCES

[1] Stanford Visualization Group. 2011-2013. DataWrangleralpha. (apr 2011-2013). Retrieved April 10, 2018 from http://vis.stanford.edu/wrangler

[2] Edwin L. Hutchins, James D. Hollan, and Donald A. Norman. 1985. Direct Manipulation Interfaces. *Human-Computer Interaction* 1, 4 (1985), 311–338.

[3] Sean Kandel, Andreas Paepcke, Joseph Hellerstein, and Jeffrey Heer. 2011. Wrangler: Interactive Visual Specification of Data Transformation Scripts. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2011), 3363–3372.

[4] Bahador Saket, Arjun Srinivasan, Eric D. Ragan, and Alex Endert. 2018. Evaluating Interactive Graphical Encodings for Data Visualization. *IEEE Transactions on Visualization and Computer Graphics* 24, 3 (2018), 1316–1330.

[5] Arvind Satyanarayan and Jeffrey Heer. 2014. Lyra: An Interactive Visualization Design Environment. *Computer Graphics Forum* 33, 3 (2014), 351–360.

[6] Ben Shneiderman. 1982. The future of interactive systems and the emergence of direct manipulation. *Behavior and Information Technology* 1, 3 (1982), 237–256.