Instructions

This version of the framework is meant to be used in the design process of process diagrams.

Before using the framework, the designer should know the purpose of the diagram as well as attributes that are planned to be visualized.

The framework is presented in an hierarchical structure, where high-level questions are divided to sub-questions. The user should answer only the last level sub-questions – the questions, which are accompanied with possible answers.

The answers are presented either as radio-buttons (select one), select-boxes (select one or several) or blank spaces to fill in. In addition, strengths and weaknesses of the answers are listed where possible. The strengths and weaknesses are in a table format, where strengths are on the left (with bullet-points marked with "+") and weaknesses on the right (with bullet-points marked with "-").

Question 1.2.2 "Which channels express the attributes?" is exceptional. It is not answered by alternative options. Instead, visual channels for ordinal and categorical attributes are listed in order of effectiveness. In addition, there is a list of common practices of the application of the visual channels on process maps. The user can use those lists as a point of inspiration.

The two first questions are answered for the user, because the current version of the framework is designed according to those pre-selected choices – the framework helps to make design-decisions about node-link diagrams (1.1.1), where nodes and links are separated from one another (1.1.2).

It is recommended to follow the proposed sequence of the framework, but it is not required.

The user does not have to answer all the questions. The questions that are not relevant to the task at hand should be left unanswered.

The output of the framework is a collection of design decisions that need to be developed further into a diagram by the user. The output of the framework is not a ready-made solution for the diagram.

The rest of the document consists of two parts – a table of contents of the framework and the framework. The table of contents shows the basic structure of the framework. The framework includes hierarchical design-questions and answers as well as strengths and weaknesses of the answers where applicable.

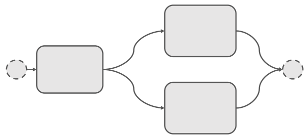
Table of Contents of the Framework

1. How to encode data?

1.1 How to arrange data?

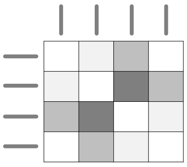
1.1.1 What is the base diagram?

✓ Node-link diagram



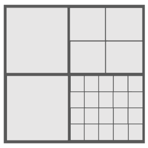
+ used for the task of understanding the topology of a process; + used for the task of discovering hierarchy of processes (tree diagrams); + intuitive.	- not easily scalable; - complex diagrams impose a cognitive overload; - occlusion.

○ Adjacency matrix



+ scalable; + used for the task of identifying activities and estimating number of links;	- unfamiliar for most users; - not possible to find multiple-link paths – not useable for topology tasks.

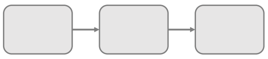
○ Enclosure



+ scalable; + used for the task of discovering hierarchy of processes; + intuitive.	- not possible to detect a sequence – not useable for topology tasks.

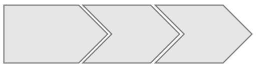
1.1.2 What are the basic elements of the diagram?

✓ Separated nodes and links



+ visualizes complex process flows, including rework; + allows drilling down to details; + better scalability than the merged version.	- requires more space than the merged version.

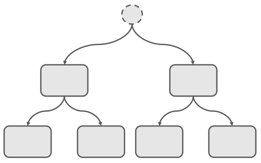
□ Merged nodes and links



+ used for high-level process flows; + little cognitive load on the user – less elements than separated version; + compact.	- not easily scalable; - usually shows only one direction of the flow.

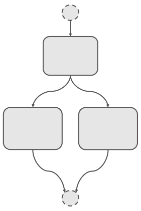
1.1.3 How are the basic elements ordered?

○ Hierarchical



+ used for the task of discovering hierarchy of processes.	- does not show the relative timing or sequence of the activities.

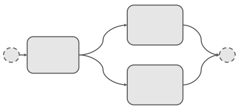
○ Sequential



+ shows relative timing of activities; + commonly used in process mining.	- detailed and coarse parts of the flow are mixed.

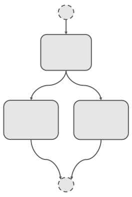
1.1.3.1 How is the sequence of the process shown?

□ Orientation of the diagram:



○ From left to right

+ intuitive in English environment – the same direction as reading text.	- difficult to scroll with a mouse.

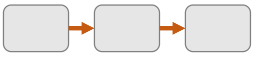


○ From up to down

+ easy to scroll.	

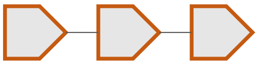
□ Directional shapes of elements:

□ Links shaped as arrows



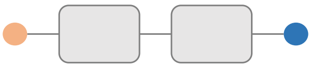
+ space-saving, while still noticeable.	

□ Nodes shaped as arrows



+ larger and more noticeable than links with arrows.	- shape channel of nodes cannot be used for anything else.

□ Other: …



□ Start and end nodes:

□ Encoding of the start node (color, shape, etc): …

□ Encoding of the end node (color, shape, etc): …

□ Other: …

1.1.4 How is the diagram aligned?

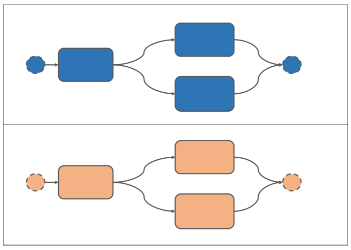
1.1.4.1 How many processes are shown?

○ One

○ Many

1.1.4.1.1 How are the process diagrams faceted?

□ Juxtaposed:

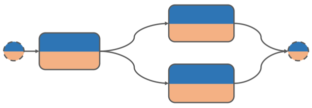


+ topology of each separate process is easy to understand.	- more cognitive load on the user than in superimposed layers when used for comparison as the eyes have to travel from one diagram to another to spot the differences.

○ Vertical

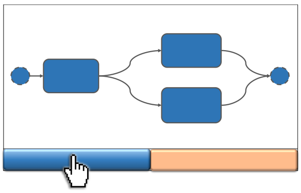
○ Horizontal

○ Matrix



□ Superimposed layers

+ easy to use for comparison purposes.	- requires attentive design of highlighting differences and other metrics; - difficult to understand topology of each separate process.



□ Separate views

+ used for faceting alternative visualizations of the same process model.	- not recommended for comparison purposes as it imposes a great cognitive load on the user memory.

□ Other: …

1.1.4.2 What is the alignment based on?

□ Best fit of proximity

+ space-saving; + easy to compute.	- sometimes proximity carries a meaning due to random chance, sometimes it is arbitrary and can lead to false conclusions.

□ Semantic meaning: …

+ uses space to represent another dimension of data.	- adds to visual clutter, especially if grouping elements, such as containment marks are included; - space-consuming; - computationally demanding.

□ Other: …

1.1.4.3 Is the layout deterministic or nondeterministic?

○ Deterministic

+ easy to reference elements based on their location.	- computationally demanding.

○ Nondeterministic

+ computationally less demanding than deterministic layout.	- the user must familiarize himself/herself with the layout after every loading.

1.2 How to map data?

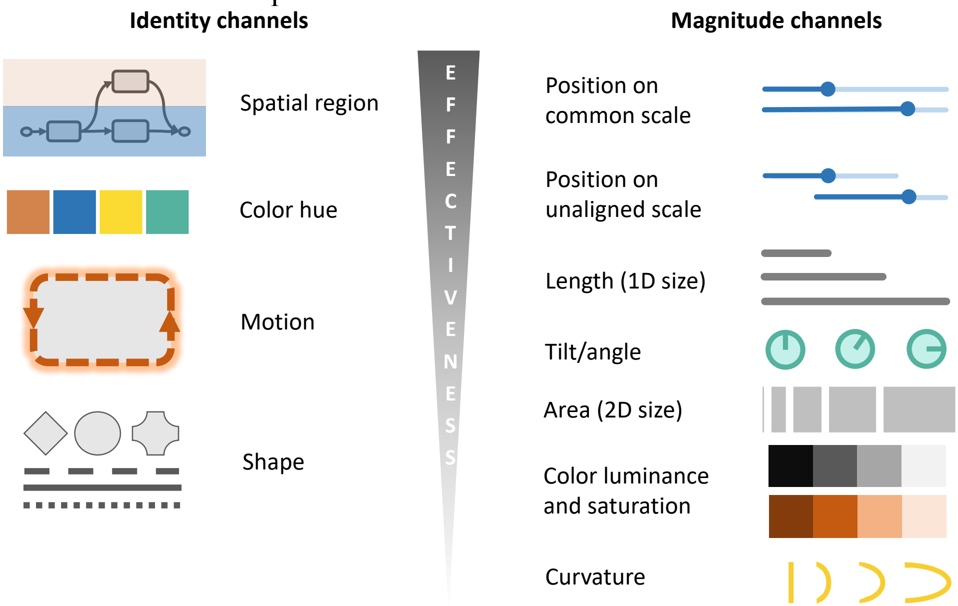
1.2.1 Which attributes are shown on the diagram?

- □ Categorical: …
- □ Ordered: …

1.2.1.1 What is the direction of ordering?

- ○ Sequential
- ○ Diverging
- ○ Cyclic

1.2.2 Which channels express the attributes?



*The most important attributes should be shown with the most effective channels (on top).
**Equally important attributes can be expressed with the same channel and the data can be faceted into exclusive layers the user can choose between.

- □ Identity channels: …

Common practices:

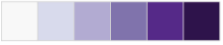
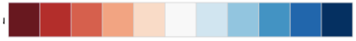
Shape	The most commonly used visual channel for categorical attributes on process diagrams because this channel is available without constraining the use of other channels. Attribute level can be communicated as follows:
	shape of nodes (circle, square, etc);
	shape of edge (continuous, dashed, etc);
	a symbol placed on a node or an edge.

Spatial region	The use of spatial region is limited due to the sequential quality of the process – ordering is already communicating the sequence

	dimension. The rest of the spatial region can be mapped in following ways (look also alignment section 1.1.4.2): vertical alignment of nodes (for horizontally oriented diagrams); horizontal alignment of nodes (for vertically oriented diagrams); adding containment marks; placing connected activities in close proximity, such as parallel activities.
Color hue *with any use of color it is important to make sure that it is visible for color blind users: http://www.color-blindness.com/coblis-color-blindness-simulator/	Color hue is often used to highlight the matches or mismatches in the process flows, when two or more processes are compared. It is a pop-out for the user to immediately identify issues. Color hue is also used in combination with color saturation for ordinal variables, when more than one performance metric is assigned to color saturation channel, e.g. shades of blue on the nodes express processing time, while shades of orange express throughput.
Motion	Used to visualize individual process instances in an animated layer (look section 1.2.3.2). It is a very strong, but underexplored channel in data visualization, which makes it prompt to misuse.

□ Magnitude channels: …

Common practices:

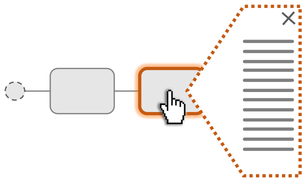
Color saturation and luminance *with the use of color saturation and luminance it is important to ensure the visibility of other elements, such as labels.	Saturation and luminance are commonly used on nodes, expressing data about activities – the darker the shade, the higher value. Color coding on nodes is stronger than on links, because it's a larger area (visible also when zoomed out). Encoding can be: sequential  diverging 
Area	Area channel is often used on links – the thicker the line, the higher the value. Nodes can be enriched by area marks when layering other types

	of diagrams on the nodes, such as pie charts.
Length	Length of links and/or nodes can show waiting and processing times. This approach offers a strong pop-out of outliers (long waiting or processing times), but it requires a lot of screen space and may not be useful for exploring the topology of the process as the diagram becomes too stretched out to get an overview.
Position on common scale/ unaligned scale	Some visualizations have dashboard diagrams (bar charts, line charts, etc) integrated into the process flow diagram to compare performance of process cohorts or activities. This can be done by placing charts on top of or next to the nodes.
Other channels	Other channels are less commonly used. Some additional channels that are not listed have also been used for showing magnitude, such as levels of blur and transparency.

□ Textual sets: …

Common practices:

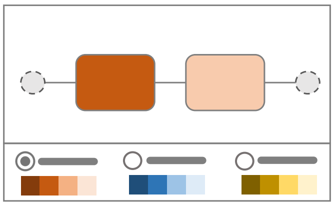
2772
events
27
activities
27h
mean
duration



Process overview statistics	Process overview statistics, such as average process time or total throughput, are usually shown in a separate area of the view, not layered on top of the diagram.
Element statistics	Various statistics of activities, such as total throughput and throughput of unique instances, are usually marked as labels and/or embedded into the diagram elements (look sections 1.2.3 and 1.2.4 for embedding and labelling).

1.2.3 How is the data faceted on the diagram?

□ Superimposed layers: …



1.2.3.1 Which channels and attributes are visible in each layer?

□ All layers: …

□ Layer 1: …, layer 2: …, …, layer n: …

1.2.3.2 Are there animated layers?

1.2.3.2.1 Which elements are shown with movement?

□ Process instance path

□ Process instance status

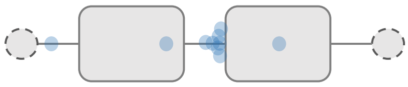
□ Other: …

1.2.3.2.2 How are the animated elements mapped?

□ Shape: …
□ Color: …
□ Size: …
□ Motion: …
□ Other: …

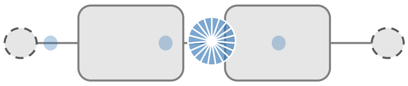
1.2.3.2.3 How to solve occlusion?

□ Transparency



+ the user can distinct separate instances.	- not possible to estimate the number of instances after the opacity level is 100% due to overlaps; - color conflicts of overlapping items, when color coding is used.

□ Merging moving items



+ better scalability than transparency.	- the user cannot easily distinct separate instances.

□ Other: …

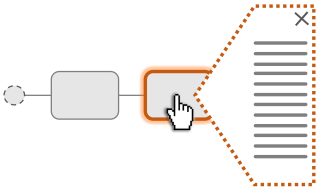
1.2.3.3 Can the user see the diagram without layers?

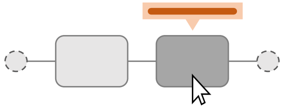
○ Yes

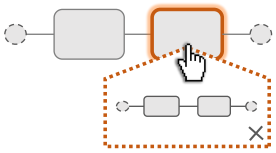
+ lessens visual distraction for topology-specific tasks.	- additional layer choice for the user – adds to the complexity of the diagram.

○ No

+ less complex set of choices.	- visual distraction for tasks that require analyzing the topology of the process







□ Embedded data: …

1.2.3.4 What is embedded?

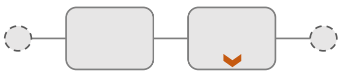
□ Attribute values
□ Labels
□ Sub-processes
□ Other: …

1.2.3.5 Where is data embedded?

□ Nodes
□ Links
□ Other: …

1.2.3.6 Is there an indicator showing the embedding point?

○ Yes



+ gives a hint of embedding to the user.	- additional elements add to the visual complexity of the process diagram.

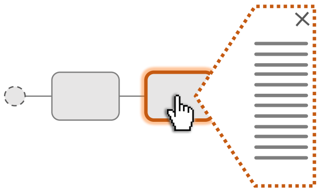
□ Shape: …
□ Color: …
□ Other: …

○ No

+ less complex diagram.	- the user has to discover the embedded data by experimenting.

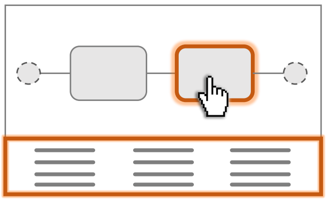
1.2.3.7 Where does the embedded data appear?

○ On the diagram

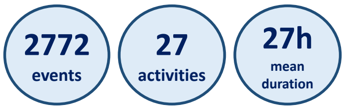


+ element and embedded data are close – easy for eyes to track.	- pop-up windows occlude parts of the base diagram.

○ Off the diagram



+ the full process is in the view when the details are shown.	- space-consuming.



□ Off the diagram: …

+ more data can be encoded into one view.	- additional sections in the view take space from the main diagram.

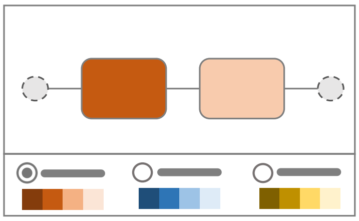
1.2.4 How does the user know the meaning of channels?

□ Legend:

1.2.4.1 Which channels and values are shown on the legend?

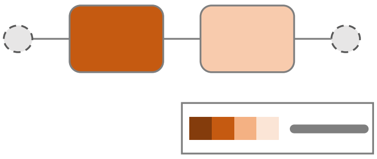
□ Channels: …
□ Values: …

1.2.4.2 Is legend separate or integrated into the control panel?



□ Integrated into the control panel

+ space-saving; + faster to use than separate area version – selecting and understanding the encoding is done as one action.	- more difficult to identify info than in a separate area version as the legend is mixed with control panel widgets.



□ Separate area

+ easy to use – a conventional way.	- space-consuming; - slower to use than integrated version – scatters user's focus between diagram, control panel and legend.

1.2.4.2.1 Is the legend dynamic or static?

○ Dynamic – includes only encoding of the selected layer

+ space-saving; + faster to identify encodings of interest than in a static version.	- works against visual memory – the user needs to understand the legend again every time it changes.

○ Static – same legend for all the layers

+ only one layout of the legend supports user's visual memory; + gives an overview of all attributes.	- space-consuming; - difficult to identify info of interest amongst many encodings.

□ Labels:

1.2.4.3 Which labels are visible?

□ All the time: …

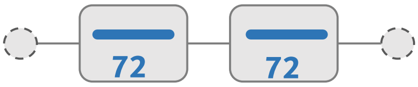
□ Layer 1: … , layer 2: … , …, layer n: …

□ Embedded (hover, click): …

□ Other: …

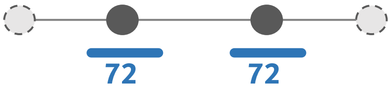
1.2.4.4 Where are the labels placed?

□ On nodes: …



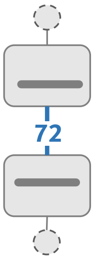
+ clearly understandable which node the label belongs to.	- the node must fit the text of the label – constrains the size and shape channel of the nodes; - the label must be visible – constrains the color channels of the nodes.

□ Next to nodes: …



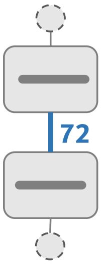
+ does not constrain the visual channels of the nodes.	- in complex processes difficult to match the label with the node; - additional elements add to the visual complexity of the process diagram.

□ On links: …



+ clearly understandable which link the label belongs to.	- the labels need additional background encoding to be visible, which occludes the links.

□ Next to links: …



+ does not occlude the links.	- in complex processes difficult to match the label with the link; - additional elements add to the visual complexity of the diagram.

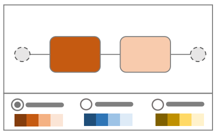
□ Other: …

1.2.4.5 How to guarantee the readability of labels?

□ Color is matched with other colors on the diagram
□ Readable size
□ Semantic zooming (look zooming section)
□ Magnified when hovered
□ Other: …

2. How to design interaction?

2.1 How can the user change the visualization?

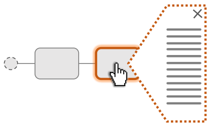


2.1.1 What can be changed on the diagram?

□ Layers:

□ Data: …

□ Encoding: …



□ Embedded data:

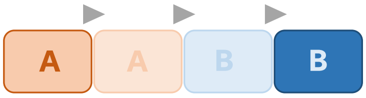
□ Data: …

□ Encoding : …

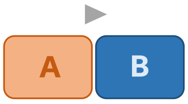
□ Other: …

2.1.2 How do the changes appear?

□ Animated transitions: …



+ keeps the connection between changed elements; + guides the focus of the user if only few elements change.	- confuses the focus of the user when many elements change; - may lead to false conclusions if the animation does not follow semantics of the data;



□ Jump cuts: …

+ quick.	- the connection between changed elements is weak.

2.1.3 What is the default appearance?

□ Basic elements: nodes, links, …

□ Layer: …

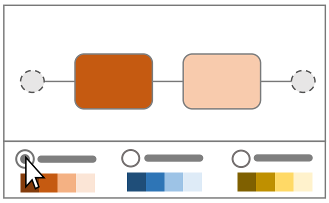
□ Embedded data: …

□ Orientation and alignment: …

□ Other: …

2.1.4 How can the changes be triggered?

2.1.4.1 Where can the user trigger the changes?

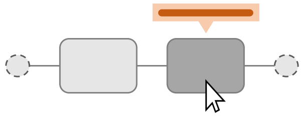


□ Control panel: …

+ gives an overview which changes can be triggered; + helps to keep track on the applied changes.	- space-consuming.

□ On the visualization: …

+ space-saving.	- triggering the changes discovered by experimenting.



2.1.4.1.1 Which actions trigger changes on the diagram?

□ Hover: …

□ Click: …

□ Double click: …

□ Drag: …

□ Scroll: …

□ Touchpad gestures: …

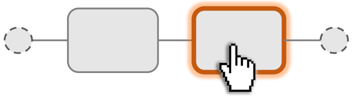
□ Other: …

□ Keyboard shortcuts: …

+ space-saving.	- triggering the changes discovered by experimenting.

□ Other:…

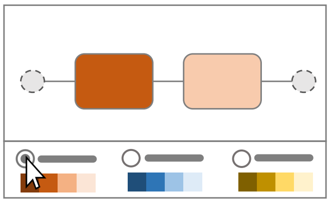
2.1.4.2 How does the user get feedback to the actions?



□ Highlight: …

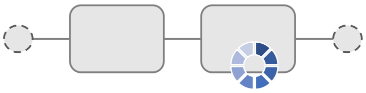
+ helps the user to evaluate if their selection matches with their intention; + used if several elements can be selected or if the element requires deselecting; + used to link data in various places on the view.	- additional elements add to the visual complexity of the diagram; - can collide with existing encoding.

□ Color: …
□ Shape: …
□ Motion: …
□ Other: …



□ Immediate change: …

+ quick if only few configurations need to be changed.	- slow if there are several configurations to be changed as every selection makes the diagram load a new version.



□ Progress indicator: …

+ used if the change takes longer than user would expect.	

□ Other: …

2.1.4.3 How can the user undo the change?

□ Deselect: …
□ Select something else: …
□ Back button: …
□ Close button: …
□ Click elsewhere: …
□ Other: …

2.2 How can the user reduce data?

2.2.1 Does the diagram need panning?

2.2.1.1 How far can the user pan?

□ Default: …
□ Up-down: …
□ Left-right: …

2.2.1.2 Which manipulation actions are for panning?

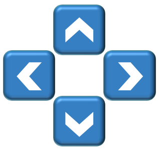
□ Scroll
□ Touchpad gestures: …
□ Keyboard arrows
□ Pinch and drag
□ Other: …

2.2.1.3 Which control elements are for panning?

□ Scrollbars

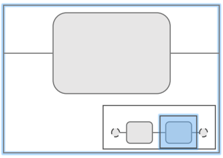
+ compact;
+ intuitive;
+ allow quick panning.	

□ Move buttons



+ compact.	- only slow (step-by-step) panning.

□ Overview-detail pane



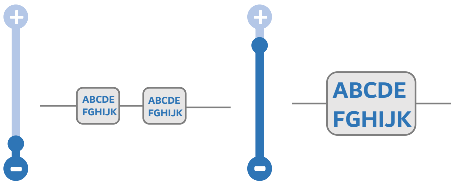
+ aids navigation in complex diagrams; + intuitive; + allows quick panning.	- space-consuming; - requires abstraction design in the overview panel.

□ Other: …

2.2.2 Does the diagram need zooming?

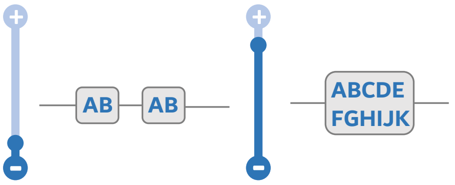
2.2.2.1 What type of zooming?

○ Geometric



+ intuitive.	- labels and visual channels lose readability when zoomed out.

○ Semantic



+ all the important elements of the diagram are visible when the diagram is zoomed out.	- additional design for elements on each level of zoom; - difficult to find a general way to shorten the activity names or other textual elements.

2.2.2.2 How close or far can the user zoom?

□ Default: …
□ The closest: …
□ The furthest: …

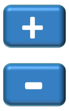
2.2.2.3 Which manipulation actions are used for zooming?

- □ Scroll
- □ Double click
- □ Touchpad gestures: …
- □ Keyboard shortcuts: …
- □ Other: …

2.2.2.4 Which control elements are for zooming?

- □ Slider

+ compact;
+ intuitive;
+ allow quick zooming.	

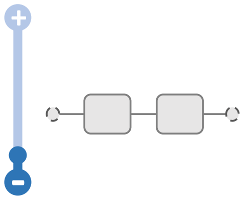


- □ Zoom buttons

+ compact.	- only slow (step-by-step) zooming.

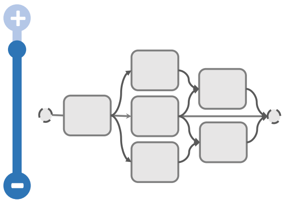
- □ Other: …

2.2.3 Does the diagram need abstracting?



2.2.3.1 What type of abstraction?

- □ Number of paths
- □ Number of activities
- □ Other: …



2.2.3.2 How simple or complex can the diagram be?

- □ Default: …
- □ Minimum number of nodes and links: …
- □ Maximum number of nodes and links: …

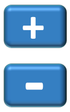
2.2.3.3 Which manipulation actions are used for abstracting?

- □ Touchpad gestures: …
- □ Keyboard shortcuts: …
- □ Other: …

2.2.3.4 Which control elements are for abstracting?

- □ Slider

+ compact;
+ intuitive;
+ allow quick abstracting.	



- □ Abstraction buttons

+ compact.	- only slow (step-by-step) abstracting.

- □ Other: …

2.2.4 Does the diagram need filtering?

2.2.4.1 Which filters can the user apply?

- □ Attributes: …
- □ Values: …

2.2.4.2 How many filters can the user apply?

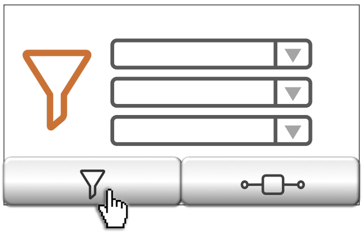
○ One

+ easy to keep track on the filters.	- does not support complex analytical tasks.

○ Many

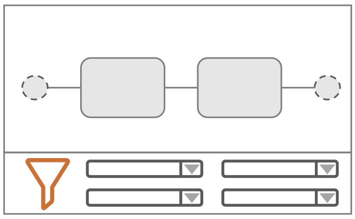
+ allows filtering for complex analytical tasks.	- requires visual aid for remembering applied filters; - computationally more complex.

2.2.4.3 Where can the user apply filters?



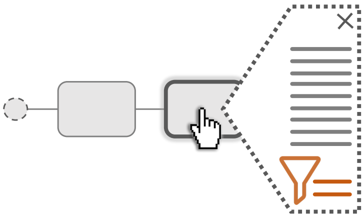
□ Separate filter view

+ used for advanced filtering as it allows enough space for all possible filtering options.	- user has to navigate to another view to apply filters; - user has to switch from process layout of the items to list layout.



□ Control panel for filtering on the diagram view

+ user does not have to navigate between views; + user can see both, process layout as well as list layout of items.	- space-consuming; - requires a concise composition of complex filters.

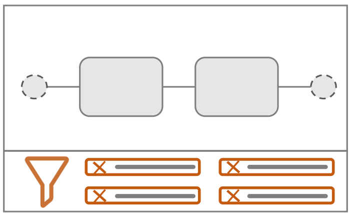


□ Shortcuts on the diagram

+ user does not have to navigate between views or control panel and diagram; + user does not have to transition from process layout to list layout of items.	- not obvious, where and how to filter as user has to find the filtering shortcuts by experimenting; - does not allow to apply complex multi-level filters.

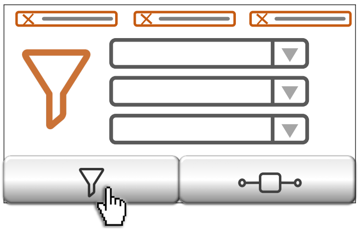
□ Other: …

2.2.4.4 How can the user keep track on the applied filters?



□ Overview on the main view

+ helps user to keep track on applied filters without any additional navigation; + user does not have to remember applied filters when using diagram view.	- space-consuming; - requires a concise composition.



□ On the filter view

+ space-saving; + applied filters do not have to be summarized concisely, but can be shown in full complexity.	- user has to navigate to another view to see the filters; - user has to remember the filters when using diagram view.

□ Other:…