Interaction History Visualization

Conference Paper · October 2012

DOI: 10.1145/2379057.2379107

3 authors:

Benedikt Schmidt
SAP Research

Sebastian Döweling
SAP Research

Max Mühlhäuser
Technical University Darmstadt

Interaction History Visualization

Benedikt Schmidt,
Sebastian Doeweling
SAP Research Darmstadt
62483 Darmstadt, Germany
firstname.lastname
@sap.com

Max Mühlhäuser
Telecooperation Group,
Technische Universität
Darmstadt
62489 Darmstadt, Germany
max@informatik.tu-
darmstadt.de

ABSTRACT

Interaction histories have been identified as a promising direction to support information workers in the execution of their work processes. However, to increase the workers' awareness about the structure of their work and to help them with the execution of their work processes, a suitable visualization is necessary. Up to now, interaction histories have typically been visualized with the classical Gantt, bar or line charts, neglecting the information contained in links between the individual items in an interaction history. Moreover, clear and empirically grounded guidance for the choice of the visualization is currently lacking. We present two graph-based visualizations for interaction histories and evaluate them against the classical visualizations in a controlled experiment. From the results, we derive a set of recommendations for the visualizations best suited for the different tasks within information workers' work processes.

Categories and Subject Descriptors

H.4.1 [**Information Systems Applications**]: Office Automation

General Terms

Human Factors

Keywords

human-computer interaction, task execution support, context, knowledge work support

1. INTRODUCTION

Information work is characterized by non-routine problem solving and a highly context-dependent execution of work processes. Moreover, frequent switches between work items are typical for this kind of work, making frequent adaptation and reevaluation of the work context necessary. As the work structure emerges ad-hoc, it is difficult for the information worker to keep track of all ongoing activities. The non-standardized execution of work processes, however, prevents the use of common workflow management systems.

Interaction histories [21] have been proposed as a promising approach to increase information workers' awareness of their own work processes, and to support them with resources relevant to their current work context. They answer typical questions like "What was the last document i worked on and where can i find it?" or "How much time did i spent reading a report?" History features in web browsers, the journal in Outlook, Social Wakoopa [8] and Rescuetime [9] are examples for this. Yet, up to now, little work has been conducted on the optimal visualization for this type of data.

Current systems often rely on the obvious choices: Gantt –, bar- and line charts. However, these visualizations neglect the information contained in links between resources used for tasks in information work (e.g. documents opened at the same time, emails sent while reviewing a presentation, etc.). Questions like "Where is the document i read while i wrote a mail to colleague yesterday morning?" cannot be answered from these visualizations. This aspect is present in graph-based visualizations; conversely, typical graph-based visualizations do not display temporal relations.

Thus, in this paper, we propose two new hybrid visualizations: the *compound graph* and the *timeline graph*. Both embed a graph visualization into a temporal one – the former focuses on the transfer of process information, using a task-centric temporal structuring of work; the latter provides a task-independent, overall perspective on the work process.

Further on, we found that there is currently no systematic evaluation of the efficiency of different visualizations for interaction histories to guide the design of interaction history based software. To address this, we use a series of tasks generated by typical questions occurring during information work activities, and evaluate both task completion time and error rate for the proposed and the established interaction history visualizations.

The rest of this paper is structured as follows: we start with a definition of interaction histories and a short summary of work in their dominant application domains; we then identify the different types of tasks information workers have to complete with regard to their work structure and derive requirements for visualizations that support these tasks, drawing upon Gestalt theory and existing work on graph understanding. Subsequently, we review related work with regard to these requirements, and, addressing the limitations, propose two new visualizations for interaction histories. We then report on the results of a first evaluation of the different

SIGDOC'12, October 3–5, 2012, Seattle, Washington, USA.

visualizations for the tasks identified before. We conclude with a set of recommendations for interaction history visualizations to inform the design of future support tools for information workers.

2. BACKGROUND

This section describes interaction histories and their relevance for information retrieval and work process awareness. The work on these aspects is fundamental to the assessment of interaction history visualization in this paper - the nature of interaction histories and the relevance for information retrieval and process awareness help to identify general tasks that need to be solved by an interaction history visualization in the next section.

2.1 Definition of interaction histories

Basically, interaction histories (see figure 1) are datasets that provide information about human activities. In the context of human-computer interaction, interaction histories refer to information about performed operations and accessed information objects.

An interaction history consists of interaction history elements. Each interaction history element stands for a logged user-system interaction, i.e. each interaction history element stands for an event that was logged when a user interacted with the system, and has temporal information about its occurrence. The granularity of the logged events depends on the used logging application. The minimum granularity focuses foreground events for information objects (e.g. when a user brings the webbrowser with the google page in the foreground, then google scholar and then Word with a document, results in the creation of three interaction history elements)[1] - typically some sort of filtering or aggregation is applied, before this information is presented to the user.


```
<event eventName="FOREGROUND_WINDOW_CHANGED" atTime="01.02.2011 16:19:26.484" eventCategory="Process"><eventattributes>
	<eventattribute name="processname" type="String" value="POWERPNT" />
	<eventattribute name="windowtitle" type="String" value="Microsoft PowerPoint - [Planning-Roundup_v3.pptx]" />
	<eventattribute name="associatedFile" type="String" value="\BaseNamedObjects\MSCTF.Shared.SFM.AFDB" />
	<eventattribute name="associatedFile" type="String" value="C:\Documents and Settings\evaluationuser3\Desktop\Work\Planning-
Roundup_v3.pptx" /></eventattributes></event>
<event eventName="FILESYSTEM_OBJECT_DELETED" atTime="01.02.2011 16:19:26.828" eventCategory="Filesystem"><eventattributes>
	<eventattribute name="name" type="String" value="C:\Documents and Settings\evaluationuser3\Desktop\Work\Planning-
Roundup_v3.pptx" />
	<eventattribute name="notifyfilter" type="String" value="FileName" /></eventattributes></event>
<event eventName="FILESYSTEM_OBJECT_CREATED" atTime="01.02.2011 16:19:26.828" eventCategory="Filesystem"><eventattributes>
	<eventattribute name="name" type="String" value="C:\Documents and Settings\evaluationuser3\Desktop\Work\Planning-
Roundup_v3.pptx" />
	<eventattribute name="notifyfilter" type="String" value="FileName" /></eventattributes></event>
<event eventName="MSPOWERPOINT_EVENT_THROWN" atTime="01.02.2011 16:19:26.968" eventCategory="Application"><eventattributes>
	<eventattribute name="title" type="String" value="Planning-Roundup_v3.pptx" />
	<eventattribute name="content" type="String" value=" Planning Demand and Master Planning Roundup Introduction Author: Andreas
Goeb&#xD;Date: 2011/02/01 Demand Planning Purpose&#xD;improve decisions affecting demand accuracy &#xD;calculation of buffer or
safety stocks &#xD;Results: &#xD;Demand Plan&#xD;Benefit:&#xD;Increased performance of each supply chain entity Master Planning
Purpose&#xD;synchronize the flow of materials along the supply chain&#xD;Range: Mid-term □ At least one seasonal
cycle&#xD;Contents:&#xD;Production&#xD;Transport&#xD;supply capacities&#xD;Seasonal stock &#xD;balancing of supply and
```


Figure 1: Excerpt from an interaction history.

2.2 Applications of interaction histories

There are two main areas of application for interaction histories: information retrieval and work processes awareness.

2.2.1 Use in information retrieval

Interaction histories support different information retrieval tasks. For these tasks, the use of interaction histories is twofold:

Collection creation.

Interaction histories are used to create collections of information objects. These collections may be input to recommender systems or useful structures to support quick manual information object access. The interaction histories may support the manual creation of such collections or the automated creation based on given criteria.

Manual creation Interaction histories can support users, when they create collections of information that belongs together. This strategy has been implemented in the UMEA system [10] and Sphere Juggler [15].

Automatic creation In contrast to the manual collection creation, a system might identify information objects in the workprocess that are relevant and that can be grouped by a certain criterion. A simple criterion is access time. The visualization of access time based collections can be found in most modern applications, e.g. in the form of "recently used files"-lists, web-browsing histories or completion proposals for search fields.

Another frequently used criterion is the task: systems try to identify information objects that belong together as they belong to the same task. This is a complex operation, as it needs operationalize relevance and relatedness and measure it for resources in possibly multitasking work processes (e.g. do a Google Scholar site and an an email accessed while a browsing this site belong to the same collection?). Systems often use textual content of accessed resources and temporal information to create information object clusters. Examples for such systems are the CAAD system [20], Swish [16] and Transparency [22].

Recommendation.

Interaction histories are also used to create proactive recommendations: based on interaction histories, the system compares a current user work context with stored information about work contexts to identify information potentially supportive to the current activity.

Such information may be collections - possibly also created by interaction histories - that seem to fit to the users' work context. This identification is often implemented using supervised machine learning – the complexity of this task lies in the demand for measures of similarity in interaction histories and the need to identify task switches. Systems that address this are the Task Tracer System [23], the aposdle monitor [14] and UICO [19].

The examples show that information retrieval support based on interaction histories has been implemented both in the form of standalone applications and as an application feature.

2.2.2 Use for work process awareness

Interaction histories give a detailed insight into work processes, which is beneficial for weakly structured and weakly formalized work. Interaction data has already been used in the 80s to improve the understanding of such work processes, e.g when Bannon analyzed users' activity organization for command lines[1].

Recently the application of interaction histories to improve information workers' understanding of their own work processes gained attention [5, 8, 9]; this improved understanding is achieved by visualizing interaction histories for the user:

The Student Activity Monitor and the Contextualized Attention Metadata Dashboard are learning analytics tools [5]:

[1]An information object is a discrete entity that carries information following a defined encoding scheme and that typically is interpreted by an application.

interaction history elements are considered as traces of attention and visualized as charts to learner and teacher to improve learning processes. Other analytics applications are Social Wakoopa [8] or Rescuetime [9].

Social Wakoopa is a social network that visualizes application- and information object usage for an individual and his/her contacts, using lists, bar and line charts. Rescuetime displays executed work in the form of bar- and line charts, and provides features to support more focused work (e.g. the system denies access to certain resources during a specified period of time). The Outlook journal offers the feature of logging Microsoft Office specific information object access (emails, tasks, presentations, spreadsheets and documents). For the logged information objects open and close times are collected and visualized in a list or a Gantt chart.

The given examples illustrate that bar-, line- and Gantt charts are currently the preferred visualization types for interaction histories.

3. REQUIREMENTS

To maximize the benefit of interaction histories in the described application areas, easy consumption and simple integration of interaction history access into work processes is crucial. A visualization of interaction histories can be considered useful in this context if it allows users to easily solve interaction history related tasks.

This section identifies typical tasks information workers solve with an interaction history visualization and discusses the mental effort for decoding the respective visualization. From these tasks (and the task-specific decoding workload) requirements for useful interaction history visualizations are deduced, and criteria to check for the requirements are identified.

3.1 Task classes for interaction history visualization

Information retrieval and improved process awareness have been presented as domains that benefit from interaction data. In the following, we present four classes of tasks that are fundamental for the application of interaction data in the named domains. The task classes are result of a review of functionalities offered in the context of interaction history visualization, discussions with information workers and a review of usage scenarios of the respective features in commercial products.

3.1.1 Task classes for interaction history related information retrieval

To use interaction histories for information retrieval, we distinguish two different types of retrieval: retrieval by description and description by relatedness in the work process:

DC Tasks (retrieve by DesCription).

Keyword-based search for an information object included in the interaction history.

Answers to: "Where can I find the document *doc* that I accessed earlier?", "I am looking for the document that is described by keywords a,b, which I accessed earlier..."

Example: The user searches an information object "sales report.docx" based on descriptive keywords.

Referred to-as: DC tasks

RO Tasks (find by RelatiOn).

The RO task is an information object search, based on the usage context of an information object. A user remembers certain objects he interacted with or a timeframe and wants to identify the related resources. This especially addresses a context based memory.

Answers to: "I search an information object I accessed earlier, but I do not know enough to find to find it by description. But I know what else I did at the same time time."

Example: A user remembers that there was an interesting document, but does not know enough to enter a description; however, he/she remembers another document while he/she wrote the first document. He/she searches for the remembered document and identifies the related document.

Referred to-as: RO tasks

3.1.2 Task classes to improve work process awareness

To use interaction histories for improving work process awareness, we consider temporal information as relevant: usage times and durations with start and end times.

UD Tasks (identify Usage Duration).

The time spent with an activity is difficult to identify for users [2]. However, an improved temporal understanding of work is useful for planning future work as well as for time reporting (e.g. relevant for agile development methods).

Answers to: "How much time was spent with an information object?"

Example: A user worked on a contract proposal named "contract.docx" for a couple of days. To settle the proposal creation, he/she needs to identify the time that was required to create the contract. He/she uses the actual working time with the document as an important hint to the total time needed.

Referred to-as: UD task

UT Tasks (identify Usage Time).

This class involves tasks, in which users identify what was done at a certain point in time, which activities followed other activities, or a sanity check whether a certain activity was actually performed.

Due to the number of accessed information objects, and complex planning and replanning processes that are involved in information work, individuals forget aspects of their work process [3]. An improved structural understanding of work helps to avoid retrospective and prospective memory failures. Retrospective memory addresses remembering what was done. Prospective memory addresses remembering what was planned to be done. Especially in the context of interruptions both memory types are crucial, as failures of both types result in higher failure rates when work processes are intended to be resumed after interruptions [4].

Answers to: "What was done in the beginning of the process?" "What was continuously relevant?" "How did the work process proceed?"

Example: A user is asked to tell a colleague, how he/she created a document. He/she needs to remember what he/she initially did and reviewed, what he/she did after that and so on. Therefore, he/she needs to get on overview of the work process.

Referred to-as: UT tasks

In the following, we will focus on the RO, UT and UC tasks. The DC task will not be investigated further, as it is a classic information retrieval task, that is not focus of this work.

3.2 Characteristics of interaction history visualization

To solve tasks with an information visualization, the required information needs to be identified in and decoded from the visualization. Although, decoding is an individual process, it follows certain regularities. In the following, general rules of visualization comprehension are discussed. The gained insight is applied to the previously identified interaction history tasks, to derive requirements for interaction history visualization.

3.2.1 Human information visualization decoding

Interaction history visualization needs to consider how visualizations are encoded by humans. Reading of visualizations is different from the reading of text. Good information visualization uses this effect and helps to understand information quicker with less errors [24]. This benefit only holds if the visualization is perceived without high mental effort. Important aspects of graph understanding are summarized in the following. The summary follows a theory of graph understanding by Pinker that relates graph understanding and mental effort [18].

Graphs communicate n-tuples of values on organized scales. Scales and values are encoded as visual objects that apply visual features to display information (length, position, lightness, shape, etc.). Graph understanding requires 1) an encoding of the physical dimensions of graphical elements and 2) an understanding of the meaning of the scales, the elements and the objects they stand for. The interplay of both aspects is crucial. A complex visualization requires high mental effort to decode the image and to identify the scales and the relation of objects to the scales. Objects that represent scales may realize a coordinate system. Based on the coordinate system, other elements are perceived and compared.

An easily consumable visualization is understood almost effortless. To realize this, perception laws need to be applied to optimize the graph drawing with respect to the visualization goals. Important perception laws are formulated by Gestalt theory: proximity, similarity, common fate, good continuation, closure, figures, ground and connectedness [7]. The laws hint to those graphical formations that are decoded almost effortless by an individual.

Pinker [18] stresses that first the spatial organization of objects (following Gestalt theory), and then trained attributes are decoded following a decoding likelihood: the unconscious decoding of spatial organization reveals objects which are decomposed into scales and values. Values are directly decoded as being relative to the scales, and as being relative to all existing values. Only in a second step, conscious processes can enhance the understanding of the graph, requiring, however, mental effort and time. Different limiting factors complicate graph understanding. Individual processing capacity is limited. Human beings can separate between four and nine elements at a time. The amount of elements is even fewer if processing resources are devoted to a concurrent task.

3.2.2 Requirements for useful interaction history visualizations

Pinkers work shows that visualizations can be optimized with respect to the mental effort required to solve a specific task.

To optimize interaction history visualization for the identified task classes, certain requirements must be met - the following list contains requirements that stem partly from the general principles for useful visualizations as discussed above, partly from several personal discussions about comprehension of activity data – on the on hand with experts in information visualization, on the other hand with users of existing tools that integrate interaction history visualizations. The term simple encoding refers to the application of Gestalt laws to simplify encoding for the specific information type:

RO-Tasks requirements:

(RQ1) Simple encoding of relations: Relations between interaction history elements should be easily identifiable, i.e. when the user has switched from one information object to another, this needs to be clearly visible.

(RQ2) Weighted relations: Relations should be weighted to display their relevance, i.e. when a user has switched frequently between two information objects, the frequency should be visible in the visualization.

(RQ3) Simple encoding of timeframes: Time should be decomposed into discrete time periods, so called timeframes, to structure interaction history data. Thus, a user can identify a certain time period like “yesterday morning” and see which activities were performed in that period.

UD-Tasks requirements:

(RQ3) also applies.

(RQ4) Simple comparability of time data: Temporal data should be associated to a scale that enables easy identification of timeframes, i.e. a user should be able to extract information like “happened before”, “happened after” or “’happened while” easily.

(RQ5) Preservation of process information: Interaction history element presentation should show how the visualized work process was structured, so that a user can assess easily that information object A was accessed in the beginning, whereas information object B was accessed towards the end of the timeframe under consideration.

UT-Tasks Requirements:

(RQ6) Simple encoding of usage times: The overall time the user accessed an information object in a specific timeframe should be easily decodable. Thus, the user can easily see how much time information object A was accessed between e.g. 4PM and 5PM yesterday.

(RQ7) Simple comparability of usage times: Usage time should be associated to object scales (following Gestalt laws) to enable a simple identification of values and direct comparison of the respective values. This way, a user can easily compare different duration times to extract "longer than" or "shorter than" information.

General Requirements:

(RQ8) Limit amount of perceptual units: The visualizations needs to be understandable for large interaction histories. As human perception capability is limited, the visualization needs to find useful ways of structuring large data sets.

(RQ9) Easy to learn: The visualization should not require extensive learning effort.

4. RELATED WORK

Different interaction history visualizations are currently used both in commercial applications and in research prototypes to solve the discussed tasks. These tools use lists, line- and bar charts, Gantt charts, or grouped object sets (as proposed by Rattenbury[20]). In the following, we will focus the discussion of the visualization requirements on bar-, line- and Gantt charts.

We will not discuss lists and grouped object sets in more detail, as the former suffers from the large cognitive effort required to decode information as the list grows, and the latter focuses on DC tasks and does not encode time or relation information beyond grouping objects according to a shared context.

Bar- and line charts.

Bar- and line charts are the dominant visualizations for interaction history-based analytics. These charts use graphical elements that allow an easy identification of value information on a coordinate system (see figure 2). Based on the coordinate system, the elements are directly relatable among another. Bar charts are especially useful to compare values against each other. Line charts are useful to identify trends in data. The visualization is well known, thus usually requires little learning. For large data sets, however, bar- and line charts become complex to read.

Bar and line charts are capable of a simple encoding of timeframes and durations in the coordinate system. The displayed shapes can be compared among another. Relations are not visualized, and can only implicitly be deduced based on information objects included in a timeframe. Thus, for RO tasks, bar- and line charts are not suitable due to the lack of relation information. For UT tasks, they are only partly suitable, as the relation between the different visualized timeframes is not encoded, which complicates the understanding of the process (e.g. once a user wants to know if an information object was used in the beginning of the reviewed time only, he/she needs to check each timeframe). For UD tasks, however, the bar- and line charts are suitable, as usage times are encoded in a way that makes them easy to decode and compare.

Examples: Social wakoopa [8], rescue time [9], the CAM dashboard [5] and the student activity monitor [5].

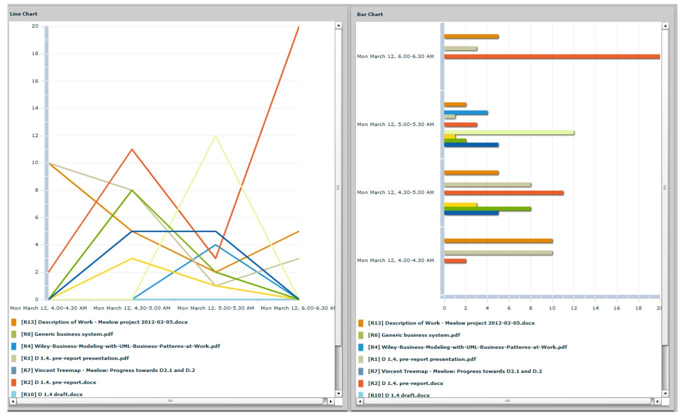

Figure 2: Bar and line chart visualization.

Gantt Charts.

When Gantt charts are used for visualization of interaction histories, this implies a (two-dimensional) coordinate system, with rows of text identifiers on the y-axis representing one or more interaction history elements, and a continuous timeline on the x-axis (see figure 3). Access durations are visualized as blocks with a start, an end and a duration that is expressed by the extent of the block. Relations among elements are visually represented by the proximity or the overlapping of blocks on the timeline. Gantt charts are well known, therefore require little learning. For huge data collections, they may, however, become overly complex.

For RO tasks, an encoding of relations is given, although it is not necessarily simple, in particular for elements with long duration. The weight of a relation is the number of similar information object successions, i.e. similar bar successions in the visualization that need to be manually counted. The identification of timeframes with gantt charts is simple, as they are simply encoded in the timeline.

The given aspects make Gantt charts a better choice for RO tasks than bar- or line charts, but they are still complex to read. For UT tasks, a simple encoding of timeframes is given. Comparison is complex, as the length of shapes at different y-positions needs to be compared. The process information is well preserved in gantt charts. One can assume that solving UT tasks with Gantt charts works well, but requires time for compare operations and search along the timeline. UD tasks are complex with Gantt charts, as the usage duration is spreaded across the timeline, or encoded in additional text, which makes the tasks solvable, but requires high metal effort.

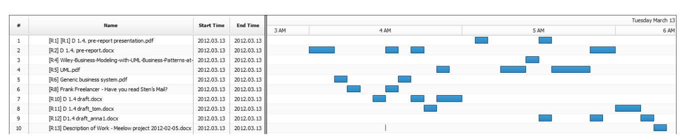

Figure 3: Gantt chart visualization.

Example: The Outlook journal provides a Gantt chart visualization of information object usage. Each information object has a dedicated row. In this row, a bar denotes the time the respective resource was open.

5. CONTRIBUTION

To our best knowledge, existing visualizations of interaction histories do not make use of graphs. This is surprising, as there is a need to display elements that have more than one predecessor and more than one successor (e.g. when interaction history elements for the same information object are aggregated). To visualize such a structure with numerous connections, graphs are useful, as they apply the law of uniform connectedness[2]. An important requirement for graphs is good readability, even when they contain many elements. [6] studies on nodes with 50 and more vertices show an increasing complexity of graph decoding and understanding.

Two specific types of graphs have gained increasing relevance [12] and are of specific importance for this paper: *dynamic graphs* address the visualization of time in graphs. Element evolution is displayed by the addition or deletion of edges and nodes, e.g. using animation. *Compound graphs* are static graphs organized based on semantic clustering, i.e. a second type of order, e.g. hierarchy or group is used to organize sets of nodes. Compound graphs are used e.g. in plate notation and UML diagrams.

In the following we shortly discuss a straight forward visualization of interaction histories with graphs, which reveals difficulties w.r.t. readability and temporal understanding of the visualized data, especially when the interaction history contains many elements. We then propose two new visualizations to address the readability and the visualization issue: the *timeline graph*, a new variant of dynamic graphs, and a *compound graph*, a variant of the compound graph applying a hierarchical structure.

5.1 Limitations of simple graphs for interaction history visualization

A straight forward graph visualization uses vertices to show interaction history elements and edges to show relations between the objects. Temporal information is added as label to nodes. This type of visualization has been integrated into a research prototype and was evaluated. 9 users used the research prototype for 2 weeks. This use was accompanied by a series of questionnaires that tracked trends in the perception of the named tool.

Most participants initially expected the graph representation of their work to be useful or very useful (6 of 9). After two weeks in which they used the tool in a normal work context, however, a number problems became apparent: the appreciation of the graph representation for the work decreased significantly (not useful (2 of 9), partly useful (2 of 9), moderately useful (2 of 9), useful (3 of 9)). 5 of 9 users considered reading the graph to be very complex. In a subsequent interview, all participants stated that they found the graph view interesting, but did not find a connection to their daily work tasks, and that it was time consuming to interact with the visualization, especially due to its size (after 8 hours of work a graph sometimes contained far more than 100 nodes, see figure 4). Also, the problem of decoding temporal information from the graph was mentioned informally by different participants.

[2]The law of uniform connectedness describes the effect that humans consider elements as related when they are connected by a visual element, e.g. a line. Palmer [17] argues that the law of uniform connectedness is the strongest of all gestalt laws.

The two visualizations presented below address in particular these concerns.

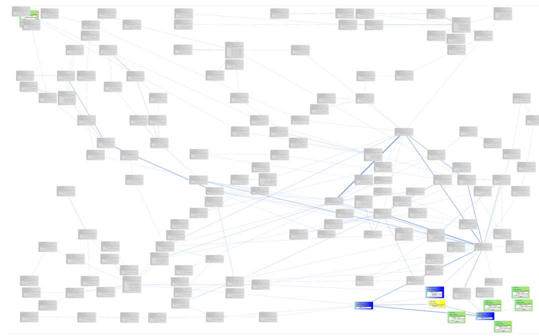

Figure 4: Interaction history visualization with a simple graph, data collected from 8 hours of work.

5.2 Timeline graph

The timeline graph addresses the central demands of RO tasks by combining temporal and relation visualization. The lower part of the visualization shows a timeline that displays the periods for which time interaction history data exists. A timeframe can be selected in the timeline to investigate in the selected period. For the selected period, the upper part displays a graph of interaction history elements. The graph encodes weighted relations by edges with different thicknesses; vertex size encodes the usage duration of the interaction history elements.

The period selected on the timeline can be moved to visualize the transformation of the visible graph by animations.

The combination of a timeline with a graph addresses the requirements for RO tasks; the encoding of a timeframe and comparability based on node size address those of UT and UD tasks. The disadvantage for UT tasks, however, is the way process information is coded in the graph: the user needs to actively change the visible timeframe to get an overview of the process. Also, UD tasks may be challenging, as the user needs to identify the timeframe that contains the information objects he/she is interested in, before they can be compared.

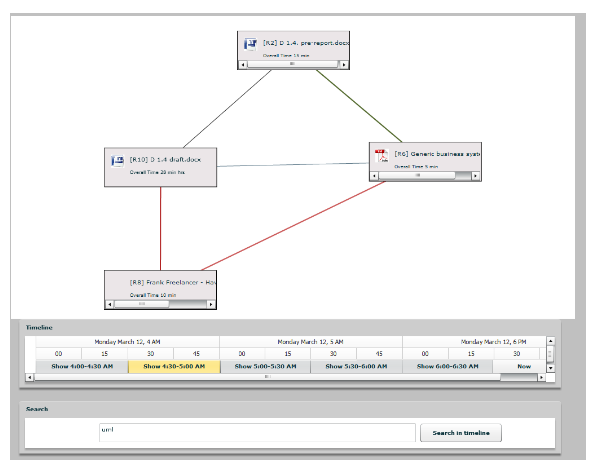

Figure 5: Timeline graph visualization.

5.3 Compound graph

The compound graph has been designed to address those requirements of UD tasks which have not been addressed completely by any of the previously discussed visualizations. In particular, the other visualizations failed to provide an easy means to transfer process knowledge. The compound graph organizes the displayed graph with respect to hierarchically ordered groups (see figure 6), i.e. elements are organized in several layers of interconnected boxes that are, in turn, embedded into a temporal coordinate system.

Each box stands for a period of time and contains a graph for interactions. An interaction history visualization is added to a box, only if all interaction history elements that is aggregated was only used during the timespan that is covered by the box and if it does not fit into the timeframe of a smaller box. The y-axis structures the duration time: the longer the box, the longer the duration. The x-axis is a timeline: the x start and end postion of the box hint to the length of the displayed period. The boxes are hierarchically ordered. The highest level contains one box with the width of the complete visualization. The level below the highest level is decomposed into two equally width boxes for two shorter periods. The lower level again, has twice the boxes, each with a width half the width of the boxes representing again shorter periods.

The hierarchical structure simplifies the process of identifying timespans of interest. The boxes restore a kind of "fuzzy process knowledge" in the visualization that structures interaction history data along questions like "What did i do in the beginning?" or "What was always relevant?". For example, in the lower left box are only those resources that were interesting only in the beginning. In the lower right box are those activities that were only performed at the end of the timeframe.

The graph fulfills all requirements for UT tasks: timeframes are encoded in a coordinate system that transfers process knowledge based on hierarchical structure. Representation of durations by vertex size enables comparability between elements. The requirements for RO tasks are also met: weighted relations are easy to decode and timeframes are clearly displayed. Only for long timeframes, the navigation of the compound graph is likely to be more complex than for the previosuly presented timeline graph. Finally, the requirements for UD tasks are met, although time comparation is presumably simpler using bar- or line charts.

6. COMPARATIVE STUDY

While the basic suitability of the respective visualizations for the identified tasks has already been analyzed, we set up a user study to substantiate the claims made.

6.1 Hypothesis and study design

We define suitability or usefulness of a visualization for a task by operation success (when a user solves a task with a visualization, the task can be solved correct, can be solved incorrect or can be considered as unsolvable with the specific visualization) and time investment. We posed three hypotheses for the suitability, focusing on the performance of the two proposed visualizations, the timeline graph and the compound graph, with respect to the other visualizations:

H1 The number of errors is lower for a) compound graph and b) timeline graph than for all other visualizations.

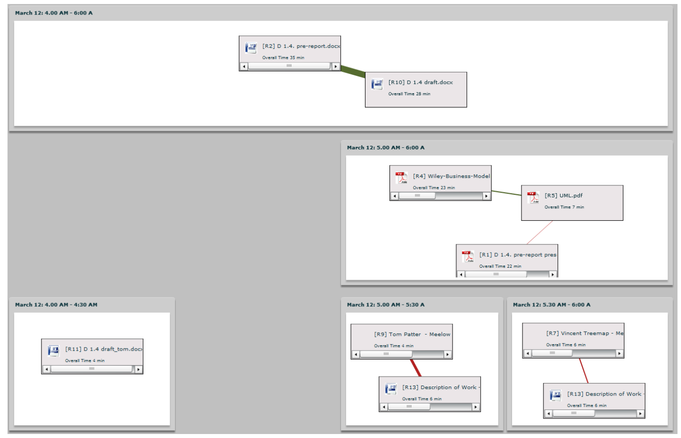

Figure 6: Compound graph visualization.

H2 Task completion time is lower for a) compound graphs and b) timeline graphs when compared to bar-, line and Gantt charts for most (at least 4) tasks.

H3 One of the graph-based visualizations outperforms the other in all tasks (in terms of task completion times and error rates).

To test these hypotheses, we created six tasks; each of them lies in one of the task classes discussed before, and each task class is represented by two different tasks (doc refers to any information object):

Task1 How much time was worked on *doc*? *(Task class: UD)*

Task2 When was *doc* used during the work process? (e.g. from 4.00-6.00 AM, from 5.00-5.30 AM, from 4.00-5-00 AM, ...)? *(Task class: UT)*

Task3 Which documents were connected to *doc* in the workprocess? *(Task class: RO)*

Task4 List the documents that were accessed only between 4.00 am and 4.30 am. *(Task class: UT)*

Task5 You have read an interesting book about patterns, when working on *doc*. Can you identify it? *(Task class: RO)*

Task6 Find 3 resources that were overall used for more than 18 min? *(Task class: UD)*

6.2 Study setup

To evaluate the performance of the visualizations on the six defined tasks, a prototypical interaction history visualization tool has been created. Initially, the tool shows a questionnaire. Then, the tool asks for the solution of the six presented tasks with the different visualizations. We have created interaction history data sets inspired by real interaction histories created during a normal working day as input for the tool.

The amount of resources has been restricted to allow an execution of the study with all 24 tests in 30 minutes. In this configuration, the visualizations display more than 7 discrete elements to assure that interaction history size is reflected in the dataset, although the scalability of the visualization with respect to very large datasets is thus not in the focus of our study. Each data set contains 13 resources, and shows 8 to 13 resources at a time. The data shows a timeframe for two hours for March 12, 4-6AM. Each task has to be executed with every visualization. To rule out learning effects, the tool randomizes among the tasks sequence, the visualization sequence and among four different data sets that are used. The different datasets share the same structure – only task completion times and information object names have been changed. The similarity among the datasets garuantees a similar complexity for the same task solved with different datasets.

To represent a realistic work process, the data sets tackle a focus topic, but also include information objects that belong to different tasks to mimic multi-tasking. The focus topic for the data sets are: 1) Software engineering/UML modeling, 2) Software engineering/Design patterns, 3) Lessing's "Hamburgische Dramaturgie", 4) Eccentric Pump Sales.

For each task, the solution provided by the user and the time spent (in ms) were logged. After the study, each participant was shortly asked to identify the visualization he/she liked the most/less.

Eleven participants were recruited for the study using convenience sampling - 10 were male, 1 female, their age was the between 25 and 60. All participants use computers heavily during their daily work processes.

6.3 Results

The initial questionnaire elaborated on interaction history use and process awareness. Ten participants stated that they use history features like timelines or history based autocompletion fields during their daily work. Four participants knew the outlook journal. With respect to memorization of work processes, 3 stated that they have problems remembering their work (2 not good, 1 okay), whereas 7 stated that they generally can remember their work processes (6 well, 1 good). Nevertheless, no one stated that he could remember all documents he worked on during the morning of the study day (study activities were all performed in the afternoon), 7 stated they remember most, 2 some of these documents – there was, however, no further inquiry to validate these reports. Only two participants stated that they spend little time with searching for documents they accessed earlier, the others spend a considerable amount of time with searching for this type of information (7 some time, 2 much time).

Number of errors.

Each participant completed 24 tasks. Each task could be solved with a correct solution, an incorrect solution, or a note that the task was not solvable with the visualization. The absolute number of errors and rejections for the tasks and visualization is visible in figure 7. It is important to note that a solution existed for each task and each visualization, although the complexity of finding it naturally differed among visualizations.

Line- and bar charts The charts show most errors (11) and most statements that a task is not solvable. Most errors and unsolvable statements occur for task 3 (9 unsolvable, 2 errors) and task 5 (3 unsolvable, 3 errors), which belong to the RO task class. This underlines the problem of visualizing relations in this chart type (they are only implicitly encoded in the timeframes). The UT and UD tasks show fewer errors, without being considerably good results. The difficulties for UT and UD relate to the complexity of bar chart reading for many elements. The comparability features disappear due to the conscious limitations.

Gantt charts The Gantt chart showed 13 unsolvable statements and 11 errors. The participants had problems with RO tasks in particular. Although relations are encoded in Gantt charts, the identification of the relations among the different rows is error-ridden, and sometimes even discarded by users due to its complexity. Considering the three task classes, the Gantt chart performed best for UD tasks, as process information is visible based on the timeline. UT tasks showed difficulties, as the users had to identify all bars for each row to identify usage times.

Timeline graph The timeline graph showed good results for all tasks with no unsolvable consideration and only five errors. The errors mainly occurred for task 4, a UD task. As the timeline graph does not include a simple encoding of the process, the participant needs to identify the work process on the period successions in the timeline which is complex and error-ridden.

Compound graph Overall, the compound graph showed the best results. Only one error and one unsolvable statement occurred for task 4.

		Task1	Task2	Task3	Task4	Task5	Task6
Barchart	unsolv	1	2	9	4	3	3
	Error	4	1	2	0	3	1
	TRUE	6	8	0	7	5	7
Gantt Chart	unsolv	0	0	5	3	3	2
	Error	1	0	5	2	3	0
	TRUE	10	11	1	5	5	9
Compound Graph	unsolv	0	0	0	1	0	0
	Error	0	0	0	1	0	0
	TRUE	11	11	11	9	11	11
Timeline Graph	Error	0	0	1	4	0	0
	TRUE	11	11	10	7	11	11

Figure 7: Correct solutions, false solutions and "unsolvable" notes (per task)

Summing up, we can confirm H1a and H1b, as the timeline graph as well as the compound graph overall showed better results with regard to the number of errors than the other visualizations.

Usage time.

All tasks were executed between 7000 ms and 120000 ms (see the scatter chart in figure 8). To make statements about the time distribution among the different visualizations we need to show significance, using an ANOVA test. This requires a normal distribution and variance homogeneity.

To test for normal distribution, the Shapiro Wilk test is applicable for a data set of the given size. During the study execution some people started to execute tasks before they understood them and spent time to think about the task. This produced outliers which were eliminated following the three sigma rule (99.73% of the values lie within 3 standard deviations of the mean) and replaced by the mean value (c.f. [13]). Shapiro Wilk shows that a normal distribution for an alpha level of 0.05 can be assumed for all but two dataset (the data for the dynamic graph in task 3 and the data

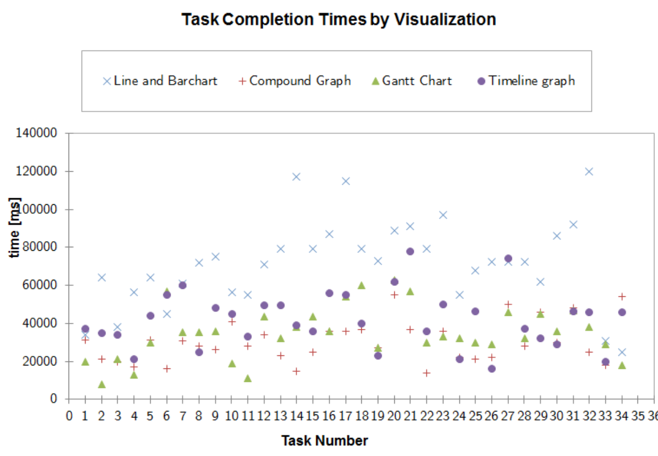

Figure 8: Task completion time (per task)

for the bar chart in task 4). We subsequently applied the Levene test for homogeneity, finding that the homogeneity is acceptable. For task 5 homogeneity assumption needs to be rejected.

As most data is normally distributed and variance homogeneity holds for all but one distributions we are allowed to apply an ANOVA test (c.f. [11]). Only task 5 was excluded, as homogeneity was rejected.

Compound graph vs. classical charts The compound graph, the bar/line chart and the gantt chart time series per task (rows) and visualizations (columns) have been used as input to a two factor ANOVA with replication. The result shows significance columns. The value for the columns is of interest as this describes the difference between the different graph types ($F(3,06) = 96{,}41$, $p < 0.001$).

Based on the average task completion times, one can investigate this further. In table 1 the average values of the task completion time per visualization and task are given. Based on this information, the strength and weaknesses of the different visualizations can be identified. It is striking that the compound graph outperforms the other visualizations (=lowest average value) for all tasks, except task 4. The identification of usage time and usage duration seems to be simple with the graph. The compound graph seems to fit the requirements for UT, UD and RO tasks very well. Only in some cases, like the UD timeframe identification of task 4, which is straight forward for the timeframe in question in that task, bar/line charts show their strength.

Summing up, we can confirm H2a.

Timeline graph vs. classical charts The timeline graph, the bar- and line chart and the gantt chart time series per task (rows) and visualizations (columns) have been used as input to a two factor ANOVA with replication. The result shows significance for columns that shows that there is significant difference between the used visualization ($F(3,94) = 13{,}41$, $p < 0.001$).

Again, we consider the average values of the time spent with each visualization for each task (see table 2): the

Task	Line and Barchart	Gannt Chart	Compound Graph
1	56446,27273	39735,53636	26446,27273
2	87247,92727	47628,09091	30801,65455
3	75305,77273	37942,14545	31446,28182
4	30008,27273	43099,16364	32181,81818
6	77801,64545	45619,82727	28454,54545

Table 1: Distribution of the average values forfor compound graph vs. classical charts in ms

Task	Line and Barchart	Gannt Chart	Dynamic Graph
1	56446,27273	39735,53636	25958,67273
2	87247,92727	47628,09091	44041,32727
3	75305,77273	37942,14545	36082,64545
4	30008,27273	43099,16364	36256,2
6	77801,64545	45619,82727	45504,12727

Table 2: Distribution of the average values for timeline graph vs. classical charts in ms

average values are better for all tasks, except task 4 and 6. These tasks ask to identify usage time and usage duration. The timeline graph performs especially good for RO tasks. Although the results for UD and UT tasks are worse, they are still convincing.

Summing up, we can confirm H2b.

Timeline graph vs. Compound graph To compare both graph visualization, the respective time series per task have been used as input to a two factor ANOVA with replication. The result shows significance for the columns. The effect of visualization types (columns) on task completion time gives: $F(3,94) = 13{,}41$, $p = 0.0004$. The null-hypothesis that the values are significantly different can be accepted. The average values of the compound graph are better than the results of the dynamic graph with timeline for all UT and UD tasks.

The graph-based visualizations show very good results with respect to task completion time and error rate for all three task classes. Still, the study does not allow a decision on one visualization which performs better for all tasks (no significant difference between the two graph-base visualizations with regard to task completion time). Therefore, H3 needs to be rejected.

Post-test interview.

After their trials, participants were asked for the visualization they appreciated the most/the least. All, except 2 considered the compound or the timeline graph as the most suitable visualization; bar- and line charts were the least appreciated visualization. This is very much in line with results given above: the graph visualizations are less prone to error, and have significantly smaller task completion times.

7. CONCLUSION

In this paper, we have discussed the challenges and requirements of interaction history visualization. Based on the application areas of interaction histories, we have identified four task classes that can be solved, using interaction histories: DC (retrieve by DesCription), RO (find by RelatiOn), UD (identify Usage Duration) and UT (identify Usage Time)). For these tasks, requirements for interaction history visualization have been specified. We have identified line charts, bar charts and Gantt charts as visualization types that are typically used to visualize interaction histories. These visualization types show weaknesses for UD and RO tasks. We have proposed two new graph-based visualizations: the timeline graph, specifically suitable for RO tasks, and the compound graph, specifically suitable for UT tasks. For all named visualization types, we have conducted a study to assess number of errors and task completion time for the execution of tasks belonging to the three classes.

Overall, the compound graph and the timeline graph have shown very good results for tasks of all three classes. Bar- and line charts and Gantt charts have shown weak results, in particular with respect to the number of errors. Bar- and line chart did, however, show good results for UT tasks, if the scale of the graph fits the searched time period. The Gantt chart showed good results for UD tasks, although the error rate was higher than for the compound graph. The best performance for RO tasks was shown by the timeline graph. The compound graph had the best overall performance, although it did not outperform the timeline graph in all task significantly.

Future work will focus on the scalability of the visualizations for very large collections of activity data. Additionally, we plan a new graph visualization which creates compounds for selected timelines. The combination of compound and timeline would combine the strengths of both graph visualizations.

8. REFERENCES

[1] L. Bannon, A. Cypher, S. Greenspan, and M. Monty. Evaluation and analysis of users' activity organization. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, number December, pages 54–57. ACM, 1983.

[2] V. Bellotti, B. Dalal, N. Good, P. Flynn, D. Bobrow, and N. Ducheneaut. What a to-do: studies of task management towards the design of a personal task list manager. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, volume 6, pages 735–742. ACM, 2004.

[3] Benedikt Schmidt, Eicke Godehardt, and Björn Pantel. Visualizing the work processSituation awareness for the knowledge worker. In *3rd IUI Workshop on Semantic Models for Adaptive Interactive Systems (SEMAIS 2012)*, 2012.

[4] M. Czerwinski, E. Horvitz, and S. Wilhite. A diary study of task switching and interruptions. *Proceedings of the SIGCHI*, 2004.

[5] E. Duval. Attention Please ! Learning Analytics for Visualization and Recommendation. *Educational Technology*, 2011.

[6] M. Ghoniem, J.-d. Fekete, and P. Castagliola. On the readability of graphs using node-link and matrix-based representations : a controlled experiment and statistical analysis. (October 2004):114–135, 2005.

[7] S. Han and G. Humphreys. Uniform connectedness and classical Gestalt principles of perceptual grouping. *Attention, Perception, & Psychophysics*, 61(4):661–674, 1999.

[8] Http://social.wakoopa.com/. Social wakoopa.

[9] Http://www.rescuetime.com/. Rescue Time.

[10] V. Kaptelinin. UMEA: translating interaction histories into project contexts. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, number 5, pages 353–360. ACM, 2003.

[11] S. Kuiper and J. Sklar. *Practicing Statistics: Guided Investigations for the Second Course*. 2012.

[12] T. V. Landesberger and A. Kuijper. Visual analysis of large graphs. *of Euro-Graphics:*, 2010.

[13] H. Liu, S. Shah, and W. Jiang. On-line outlier detection and data cleaning. *Computers & Chemical Engineering*, 28(9):1635–1647, Aug. 2004.

[14] R. Lokaiczyk and M. Goertz. Extending Low Level Context Events by Data Aggregation. *Proceedings of I-KNOW 08 and I-MEDIA '08*, pages 118–125, 2008.

[15] R. Morteo, V. Gonzalez, J. Favela, and G. Mark. Sphere juggler: fast context retrieval in support of working spheres. *Proceedings of the Fifth Mexican International Conference in Computer Science, 2004. ENC 2004.*, pages 361–367, 2004.

[16] N. Oliver, G. Smith, C. Thakkar, and A. Surendran. SWISH: semantic analysis of window titles and switching history. In *Proceedings of the 11th international conference on Intelligent user interfaces*, pages 201–209. ACM, 2006.

[17] S. Palmer and I. Rock. Rethinking perceptual organization: The role of uniform connectedness. *Psychonomic Bulletin & Review*, 1(1):29—-55, 1994.

[18] S. Pinker. A Theory of Graph Comprehension. In *Artificial intelligence and the future of testing*, pages 73–126. Lawrence Erlbaum Associates, 1990.

[19] A. S. Rath and S. N. Lindstaedt. UICO: An ontology-based user interaction context model for Automatic Task Detection on the Computer Desktop. (Ciao), 2009.

[20] T. Rattenbury and J. Canny. CAAD: an automatic task support system. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 696–706. ACM, 2007.

[21] B. Schmidt and E. Godehardt. Interaction Data Management. In *Knowlege-Based and Intelligent Information and Engineering Systems*. Springer, 2011.

[22] B. Schmidt, J. Kastl, T. Stoitsev, and M. Mühlhäuser. Hierarchical Task Instance Mining in Interaction Histories. In *Proceedings of the 29th annual international conference on Design of communication (SIGDOC)*. ACM, 2011.

[23] J. Shen, L. Li, and T. Dietterich. Real-time detection of task switches of desktop users. In *Proceedings of the International Joint Conferences on Artificial Intelligence*, volume 7, pages 2868–2873, 2007.

[24] C. Ware. *Information visualization: perception for design*. Morgan Kaufmann, 2004.