Timeline Trees: Visualizing Sequences of Transactions in Information Hierarchies

Michael Burch
University of Trier
burchm@uni-trier.de

Fabian Beck
University of Trier
fabian@fbeck.com

Stephan Diehl
University of Trier
diehl@uni-trier.de

ABSTRACT

In many applications transactions between the elements of an information hierarchy occur over time. For example, the product offers of a department store can be organized into product groups and subgroups to form an information hierarchy. A market basket consisting of the products bought by a customer forms a transaction. Market baskets of one or more customers can be ordered by time into a sequence of transactions. Each item in a transaction is associated with a measure, for example, the amount paid for a product.

In this paper we present a novel method for visualizing sequences of these kinds of transactions in information hierarchies. It uses a tree layout to draw the hierarchy and a timeline to represent progression of transactions in the hierarchy. We have developed several interaction techniques that allow the users to explore the data. Smooth animations help them to track the transitions between views. The usefulness of the approach is illustrated by examples from several very different application domains.

Categories and Subject Descriptors

H.5 [**Information Interfaces and Presentation**]: Miscellaneous

General Terms

Algorithms, design, human factors.

Keywords

Visualization, hierarchy, time.

1. INTRODUCTION

The visualization of hierarchical data is at the heart of information visualization. Information hierarchies exist in many application domains: hierarchical organization of companies, news topics and subtopics, file/directory systems, products and product groups of a department store, evolutionary or phylogenetic trees in biology.

There has been a lot of work on visualizing such hierarchies using node-link diagrams [12], radial [16], or space-filling techniques like Treemaps [7], Information Slices [1], or Sunburst [14].

In the application domains mentioned above, there are also relations between elements in the hierarchy. For example, employees are related if they communicate with each other, topics are related if they are covered in the daily newscast, files are related if they are changed simultaneously by the same person, or products are related if they are bought by the same customer at the same time. Through these relations the participating elements together form a transaction.

So far, only few researchers have developed methods to visualize such transactions between elements of a hierarchy [10, 3, 17, 2].

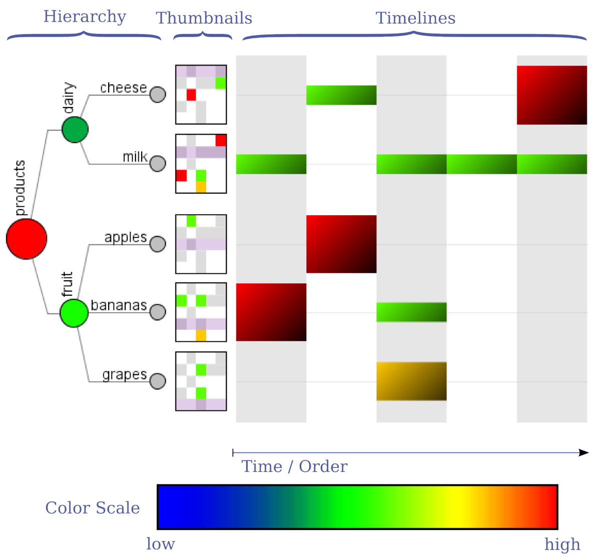

Figure 1: Visualization of the market basket example (see Table 1) as a Timeline Tree and additional explanations.

Often, we are not interested in a single transaction, but in a sequence of transactions that occur over time. Furthermore, the elements can be affected by or involved in a transaction to different extent. To model this we associate a measure with each transaction mapping each element of the transaction to a positive real number. Thus, in the application domains mentioned above the conversational partners, the selection of topics, the files, and the products bought are the elements of transactions, while the duration of the communication, the extent of coverage of each topic, the size of the modification of each file, the amount paid for a product are the associated time-varying measures.

The Timeline Tree approach presented in this paper supports the visualization of such sequences of transactions in information hierarchies in a single diagram by integrating three views (see Figure 1):

Information Hierarchy: It shows the whole hierarchy to an interactively selectable level. By clicking at a node that is currently displayed as a leaf, it is expanded; by clicking at an intermediate node, the subtree starting at that node is collapsed. Expanding or collapsing subtrees of the hierarchy can help to detect relations at different levels of abstraction.

Timelines: The sequence of transactions is visualized on a timeline drawn as an extension to the interactive tree. The elements of a transaction are represented by boxes, that are colored and sized according to the defined measure. Together with some alternative views and further features, that are introduced further below, the timeline visualization provides an extensive tool to explore and analyze the transactions.

Thumbnails: Small representations of the timeline view at each leaf node or at each collapsed node of the hierarchy enable the user to detect dependencies in which the element(s) represented by that node are involved.

We have developed several interaction techniques that allow the users to explore the data. Smooth animations help them to track the transitions between views.

To the best of our knowledge Timeline Trees is the first approach that allows users to visually explore transactions in information hierarchies. The user can analyze the evolution of transactions, the roles of their member elements, and detect when and how strong elements of the hierarchy are related.

The rest of this paper is organized as follows: In Section 2 we give a formal model of the kind of data that can be visualized by Timeline Trees. Next, we introduce the main features of our technique in Section 3 by looking at a simple example. Then, in Section 4 we illustrate the usefulness of these features by looking at data sets from various application domains. Related work is discussed in Section 5, and finally, Section 6 gives some conclusions.

2. DATA MODEL

For the purposes of this paper we model an information hierarchy as a tree where the leaf nodes represent some pieces of information, that we call items in the rest of this paper. Let $T = (V, E)$ be such a tree where V is a set of nodes, E a set of directed edges and $I \subset V$ the set of leaf nodes.

Furthermore let (μ_n) be a sequence of measures for $n \in \mathbb{N}$ where $\mu_i : 2^I \to \mathbb{R}_0^+$, $1 \leq i \leq n$, is an arbitrary measure defined on the set of items [4]. So we can model a transaction t_i for $1 \leq i \leq n$ as a set of items: $t_i := \{v \in I \mid \mu_i(\{v\}) > 0\}$.

We define a function $\mathit{front} : V \to 2^I$, that maps a node $v \in V$ on the set of its reachable leaf nodes $I' \subset I$, and functions $\hat{\mu}_i(v) := \mu_i(\mathit{front}(v))$, $1 \leq i \leq n$, that extend the measures to arbitrary nodes $v \in V$.

Also note, that Timeline Trees focus on visualizing static hierarchies, nevertheless minor changes in the hierarchy can be simply transformed to fit our data model. For instance, if an item moves in the hierarchy, it will be displayed in both positions and handled like two different items.

3. FEATURES

Table 1: Example: Market baskets.

Day	Market basket and money spent
Monday:	milk \$1, bananas \$3
Tuesday:	cheese \$1, apples \$3
Wednesday:	milk \$1, bananas \$1, grapes \$2
Thursday:	milk \$1
Friday:	milk \$1, cheese \$3

To illustrate the features of our visualization technique we use the small data set given in Table 1. It shows the market baskets of five subsequent days, i.e. the products and prices of each product that a person bought. In our example, we use the price as the measure.

For example, the third transaction corresponding to Wednesday is $t_3 = \{\mathsf{milk}, \mathsf{bananas}, \mathsf{grapes}\}$ where the value of the measure for grapes is $\mu_3(\{\mathsf{grapes}\}) = 2$. Figure 1 shows the market basket example in the Timeline Tree visualization. The visualization is composed of three views: the information hierarchy, the thumbnails, and the timelines. We discuss each of these views below.

3.1 View: Information Hierarchy

We start describing the visualization at the tree diagram of the information hierarchy on the left side. It uses a customary node-link representation where the size and color of a node encode the number of items that are descendants of the node, so one can identify major nodes even if they are collapsed.

Actually, the most important interaction functions of the tree diagram are collapsing and expanding of nodes with smooth transitions. This enables the user to explore larger information hierarchies without loosing focus and to compare data on different levels of abstraction.

The goal of the tree layout is to efficiently display the tree with labeled nodes and to let subtrees visually emerge. The former goal is realized by using more horizontal space with increasing depth of the nodes, so the tree gets more space-filling, and by adapting the label's orientation and size to the available space. The nearly orthogonal layout and smaller vertical distances between siblings help to reach the latter goal. Furthermore tooltips provide detailed information about the nodes.

3.2 View: Timelines

Timeline Trees visualize a sequence of transactions as sets of boxes, that are ordered from left to right on a diagram we refer to as a 'timeline'—in many applications time provides a natural order on the transactions. Each box represents one member element of a transaction and is positioned in the same column as the other members of this transaction and in the row of the according item.

The measure $\mu_i(\{v\})$ of an item v in transaction t_i is redundantly encoded by color and height of the box, whereas its width is fixed. So the size of a box increases linearly with the measure. But the user can also switch to fixed heights because in some applications the 'importance' of an element does not correlate with the measure.

We included numerous predefined color scales for color coding such that the user can select a suitable color scale for the task at hand. Discerning two adjacent, similarly colored boxes might be difficult, so we use a brightness gradient as a kind of cushion effect [15].

So far we discussed how to draw boxes for single items. Next we look at how to handle collapsed nodes. For that we use the function $\hat{\mu}_i$ which is defined for each node as the sum of the measure values of all leaf nodes reachable from this node (see Section 2). In our visualization this sum can be either drawn as a single box or as several vertically stacked boxes, each representing a single item. Both modes can be useful for different applications (see Section 4).

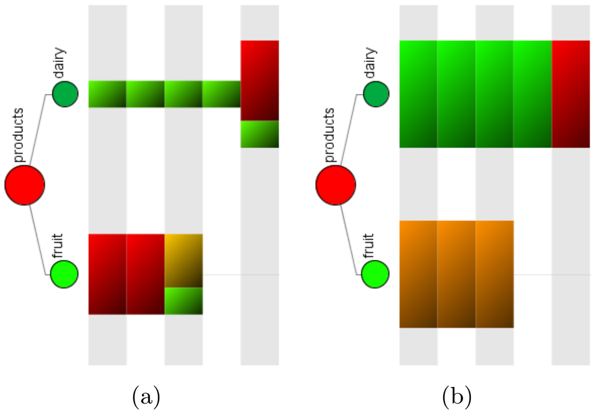

Figure 2: Market basket with collapsed nodes 'dairy' and 'fruit' in different modes: (a) height represents the measure, collapsed items are stacked; (b) unified heights, summed measure values for collapsed items.

During the interactive exploration process of a data set, a good orientation and an easy access to additional information is very important. Our visualization supports these aspects by highlighting the row and column marked by the current position of the mouse cursor and by detailed tooltip texts as shown in Figure 3. Another very useful feature is the masking of transactions. To this end, the user can select some items or collapsed nodes to form a mask set M. Only those transactions $T_M = \{t_i | t_i \cap M = M\}$ that match the mask set will be shown. All transactions that do not contain all nodes of the mask set are faded out. As a result, the user can focus on the relations between the nodes in the mask set.

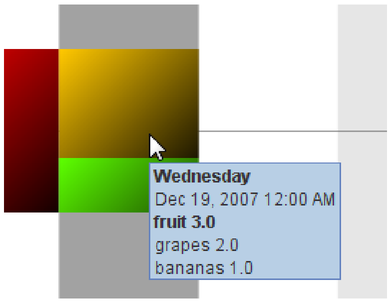

Figure 3: Tooltip for the collapsed 'fruit' node in the Wednesday transaction of the market basket example (see also Figure 2).

3.3 View: Thumbnails

The idea of masking transactions is extended by the thumbnail views of the timeline diagram. These thumbnails are displayed for every item or collapsed node at the right side of the tree diagram. They show the transactions from the perspective of the according node as if this node would be the only element of the mask set. In other words, only those transactions the node is member of are represented in the thumbnail using the selected color code, the remaining transactions are only drawn as gray boxes. As for the general mask set, the thumbnails are a good tool for identifying correlations between nodes, but in contrast to the mask set, the thumbnails are simultaneously shown for each item or collapsed nodes.

To assist orientation in the thumbnails, within a thumbnail the row of the node related to the thumbnail is highlighted as a slightly colored line. Furthermore, to countervail the disadvantage of the thumbnails' relatively small size, we implemented a magnification lens functionality that enlarges parts of the thumbnails when the mouse cursor moves over them.

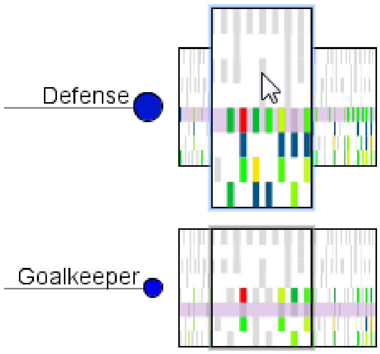

Figure 4: Thumbnail example with lens function whereas the mouse cursor is over the 'Defense' thumbnail (detailed view of the soccer match visualization presented in Section 4.1).

3.4 Alternative Representation: Time Bars

In addition to the visualization discussed above, Timeline Trees include an alternative representation of the transactions which is shown in Figure 5 for the market basket example. Here, the time or order of transactions is encoded using color coding and the measure is represented by the width of the boxes instead of their height. The boxes are drawn from left to right attached to each other, instead of positioning them in separate columns as in the Timeline representation. Thus, boxes related to the same transaction are no longer in the same column, but they have the same color. As the resulting representation is very similar to a bar chart, we call it Time Bars.

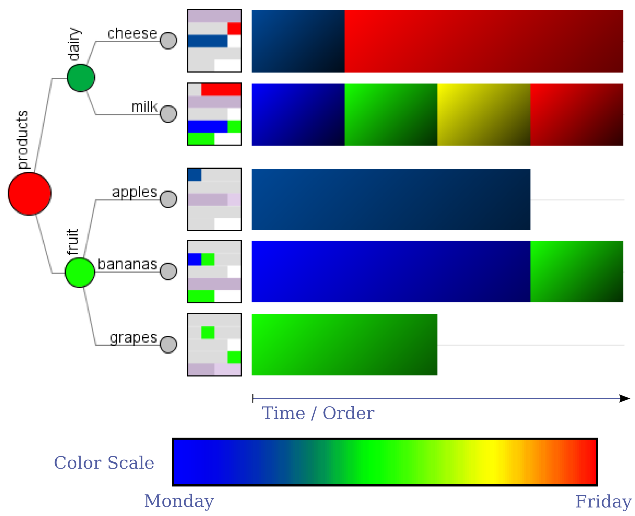

Figure 5: Visualization of the market basket example in Time Bars view.

We use Time Bars only in addition to the default visualization because the color coding of time is not so intuitive, and discerning transactions in time is not accurate enough. But for many analyses it provides the following advantages:

- The Time Bars form a kind of bar diagram that represents the aggregated measures of the items and currently collapsed nodes. So for example, one can easily detect which node is the most 'active' one.
- The shape of the diagram is much more memorable and one establishes a sort of mental map while exploring the data. Thus the orientation in the diagram and especially in the thumbnails is significantly better.
- The distribution of colors gives a more holistic overview of the temporal progress of the transactions: One can detect differences at first sight.

4. APPLICATIONS

To illustrate the features of our visualization system, we apply the system to data sets of very different domains.

4.1 Team Play in a Soccer Match

Soccer teams are hierarchically organized. Eleven players belong to each team and are subdivided into different team parts: the goalkeeper, the defense, the midfield and the offense. Additionally, players have their specific location or area on the soccer ground where they act. The number of contacts with the ball and the different players belonging to a move of the match can be seen as a transaction where each element has a measure namely the number of contacts.

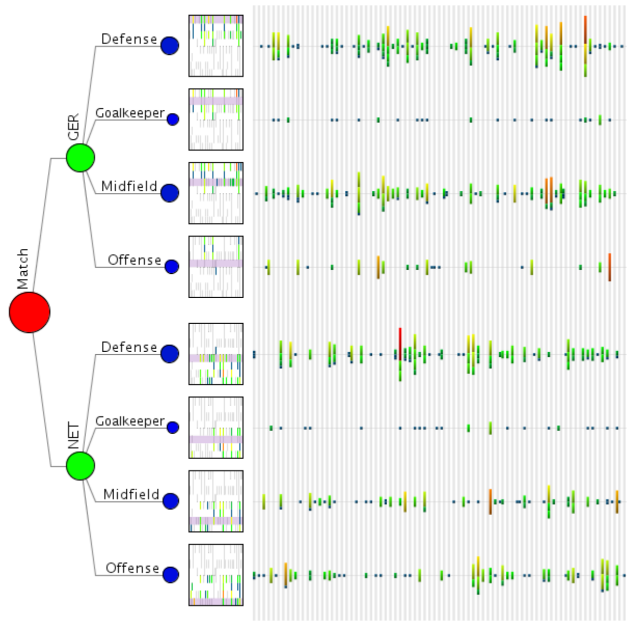

Figure 6: Timeline Trees for the soccer match between Germany and the Netherlands in World Cup Championships 1990 in Italy on team part level.

Figure 6 shows the moves of the first half of a soccer match[1]. In this visualization the organizational structure of a soccer team in terms of offense, defense, midfield and the individual players forms the hierarchy and is represented as a node-link diagram at the left hand side. Players are related to each other, if they take part in the same move which can be observed by the thumbnail view in each of the small boxes. Here, we define move as the time period during which a team has the exclusive ball possession. As the measure of a move we use the number of passes, i.e. how often the ball was passed from one member of the team to another member of the same team. Many ball contacts are indicated by high bars and red color, whereas a green color stands for a little contacts in a move, yellow is a value in between.

In Figure 6 we can also make very interesting observations about the first half of the match. The hierarchy is expanded to the level of team parts. Both defenses are the parts with the most ball contacts. The goalkeepers have only very little contacts, which is an absolutely normal phenomenon. The German offense acts not as much as their counterpart from the Netherlands. But the German midfield takes this part and therefore has much more ball contacts than the one of the Netherlands. A closer look at the lowest timeline in this figure reveals that the offense of the Netherlands increases

[1] Germany vs. Netherlands (2:1) at the World Championship 1990 in Italy

their number of ball contacts towards the end of this first half.

In Figure 7 the German midfield and offense as well as the defense of the Netherlands is expanded. The thumbnail view can give us the information that there is one transaction in which one player of each team is involved. Frank Rijkaard and Rudi Völler both received the red card and are ejected from the game. This detail on demand information can be requested by a tooltip when moving to the position of one of the corresponding bars. After this 21st minute of the first half, the following observation can be made. Frank Rijkaard was a defending player and it can be expected that the other players belonging to the defense have to do the work of the missing player. And in fact this is true. The players Adrie van Tiggelen and Ronald Koeman have much more ball contacts than before this 21st minute. Another observation is that the ball contacts of the whole offense part of the Netherlands increase right after this 21st minute and naturally the defense of Germany in the same way.

4.2 Software Evolution

Open Source software systems under version control can be used to gain interesting insights of the development process of the software. One important observation can be which files have been changed together to what extent. Furthermore it can be referred which files have been developed in which period of time. These facts can be very helpful to support software developers during the evolution process of their current project.

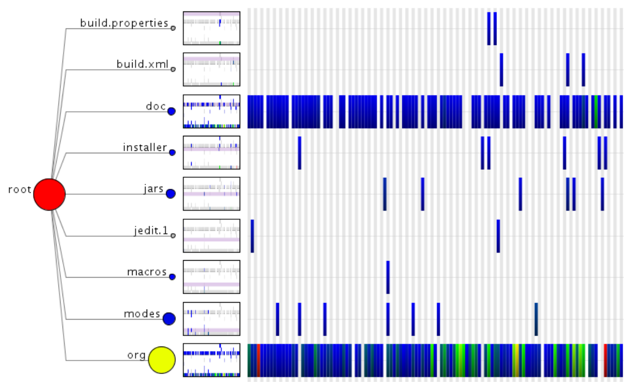

Figure 8: Transactions of a part of the JEDIT Open Source software project.

Figure 8 shows the Timeline Tree visualization for a time period of the development of the JEDIT [13] Open Source software project. In this figure, the two overall blue colored lines indicate that two software artifacts are in the center of the evolution process. The upper one corresponds to the `doc` subdirectory and the one in the lower part represents the whole source code subdirectory of the project.

Most of the transactions contain at least one file of the source code subdirectory. Documentation and source code are changed together very frequently. This can be a hint that developers almost always document their changes immediately.

A closer look at the selection of transactions by the mask set in Figure 9 that contains both documentation files `TODO.txt` and `CHANGES.txt` reveals that in nearly each case when a developer changes the file `CHANGES.txt` he also changes the file `TODO.txt`. The inverse only holds in about 50 percent of the transactions. Our hypothesis is that if someone makes a change to the `CHANGES.txt` file he always has to adjust the `TODO.txt` file because the change solves a problem or implements a feature contained in the to-do list.

4.3 World's Export

Using Time Bars instead of timelines our visualization can be used as an augmented bar chart diagram. The bars are generated by stacking the boxes of each time interval. Additionally to the conventional approach, the single bars are colored with respect to their corresponding time interval. This approach can help to observe in which time interval a bar grows more rapidly than others.

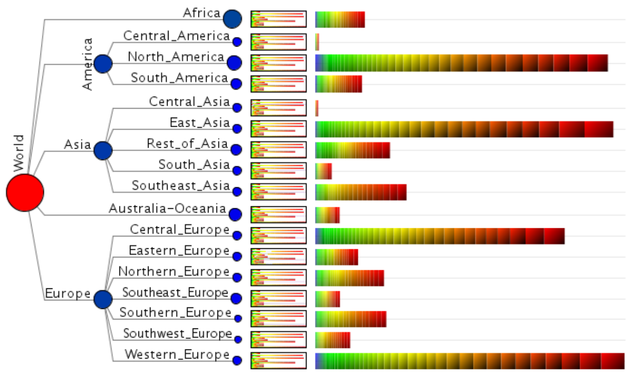

Figure 10: Export data (in Dollar) of the world's regions in Time Bars view from 1948-2005.

Figure 10 shows the yearly export data in terms of dollars for the whole world from 1948 to 2005. The year of a transaction is indicated by color, where blue indicates older transactions and red indicates more recent ones. Green, yellow and orange colors are in between. We can immediately see that Western Europe has the biggest export value for this time interval followed by East Asia, North America and Central Europe. The hierarchy can be expanded to the country level to gain insights about the export data of each country of the world. Another interesting observation is that the whole continent of Africa exports less than Southern Europe for example.

The steady growth of the single boxes from left to right is largely caused by inflation.

4.4 Movies

For this application we looked at the starring actors of movies. For each movie the set of actors is regarded as a transaction, and all movies can be chronologically ordered.

Figure 11 shows the starring actors in movies directed by Martin Scorsese. Actors are hierarchically organized according to the number of their nominations and Academy Awards. Figure 11 is divided into three parts. The left

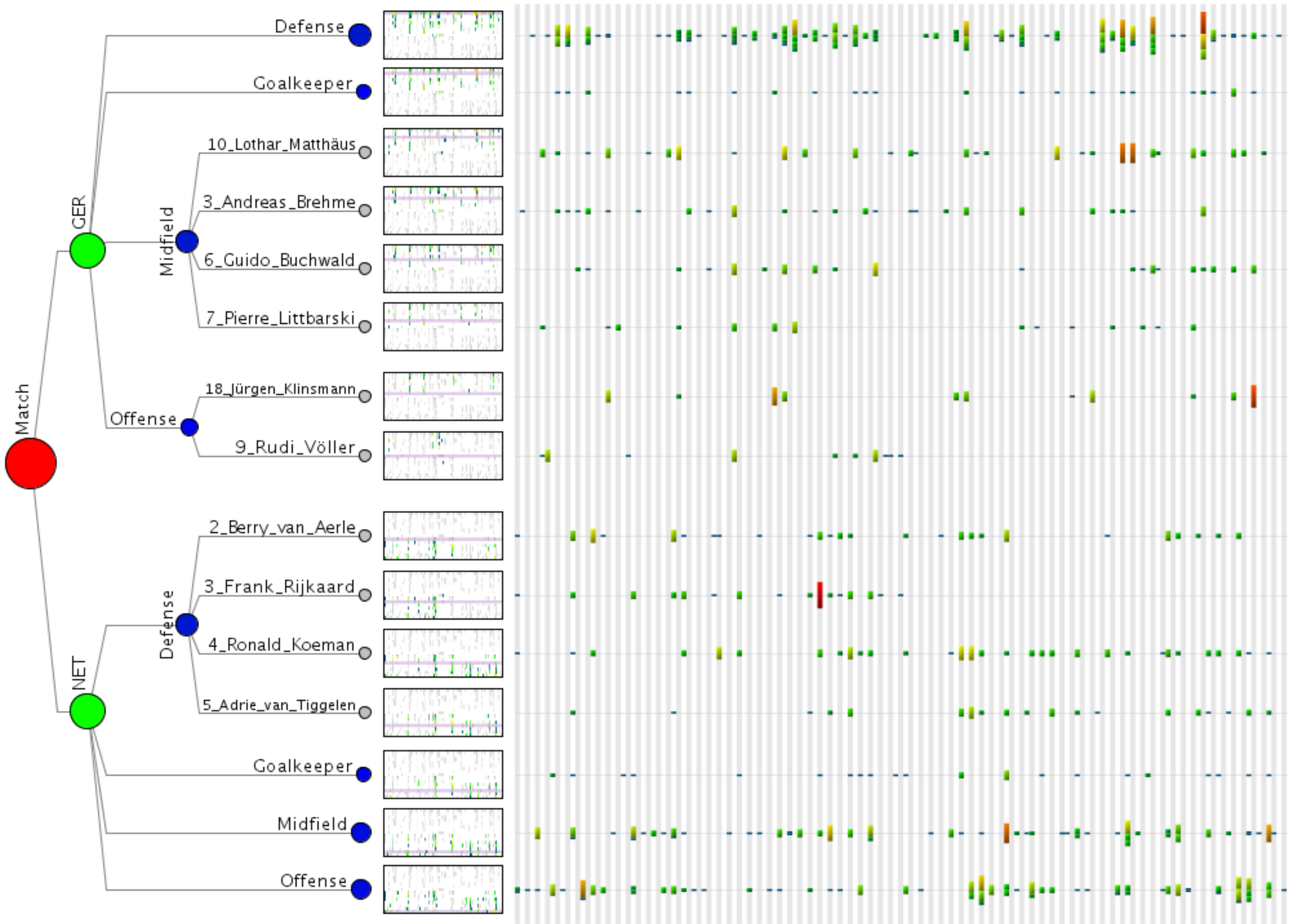

Figure 7: Timeline Trees for the soccer match with expanded team part subhierarchies.

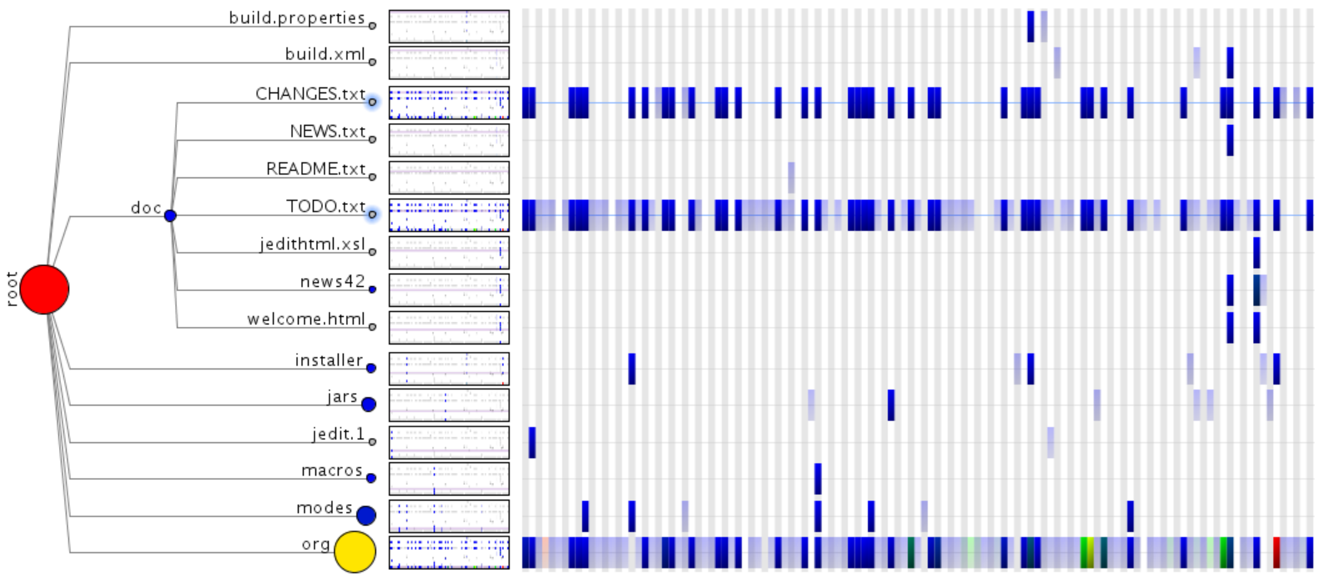

Figure 9: Timeline Trees with two files in the mask set, common transactions of the masked files are highlighted.

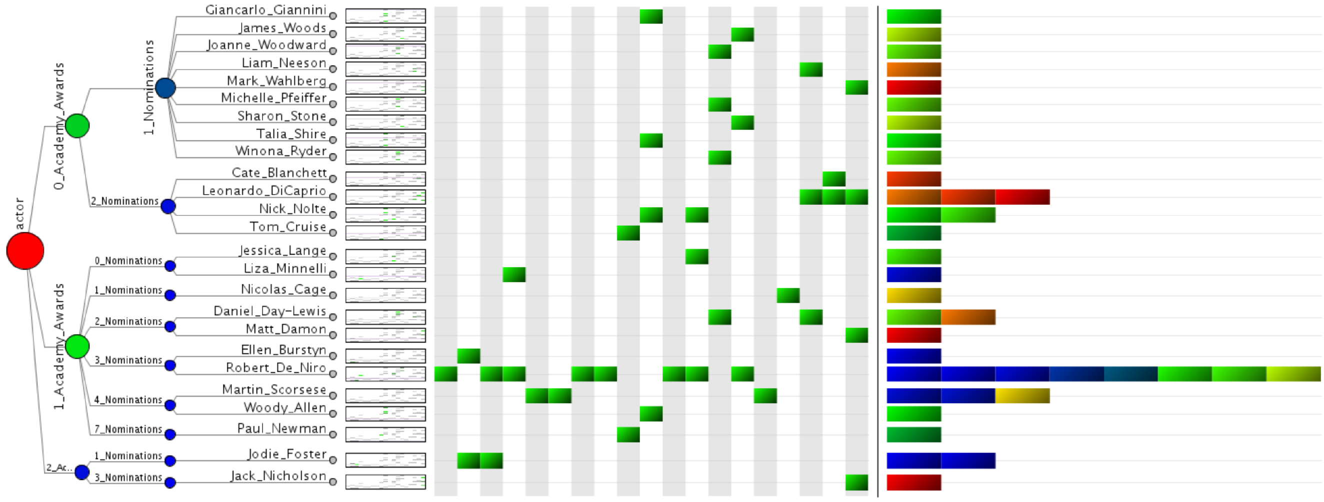

Figure 11: Movies directed by Martin Scorsese (as transactions) and starring actors (as items), that are classified by Academy Award wins and nomimations.

one shows the hierarchy, in the middle part the transactions are represented as well as the thumbnails and the right part shows the Time Bars view. The longest Time Bar is related to Robert de Niro. Many of the nominated actors only starred in one movie directed by Martin Scorsese. Only Scorsese himself and Leonardo Di Caprio appear three times. A closer look at the Time Bars or the timelines reveals that Robert de Niro was a starring actor in the past, whereas Leonardo di Caprio was a starring actor in the last three movies. The thumbnail view can be used to make observations about costarring actors. Only thumbnails of Nicolas Cage and Martin Scorsese don't have any colored elements. This means they have never been costarring with any other nominees or awardees in movies directed by Martin Scorsese. Furthermore, only actors with at least one Academy Award or no nomination at all played in Scorsese's first eight movies.

5. RELATED WORK

As Timeline Trees integrate views for hierarchies, relations and chronological data, we discuss related work with respect to these.

5.1 Visualization of Hierarchies

Information hierarchies can be seen as a special kind of graphs, namely trees. As a result, information hierarchies can be visualized as node-link diagrams using specialized graph layout algorithms [12], but also space-filling techniques like Treemaps [7], Information Slices [1], or Sunburst [14] have been developed. In Timeline Trees we use a node-link diagram to visualize the hierarchy.

5.2 Visualization of Relations in Hierarchies

Many approaches try to encode existing relations between objects as directed or undirected edges in node-link diagrams. The appearance of edges, for example, their color, shape, orientation, thickness or connection can represent a measure, e.g. the strength of a relation.

There are several approaches that extend the Treemap approach [7] to also show different kinds of relations [3, 17, 2]. For example, ARCTREES [10] is an interactive visualization tool for hierarchical and non-hierarchical relations. It extends the hierarchical view of the Treemap approach with arc diagrams to present relations. In Timeline Trees items are related by transactions. Transactions and the related measure are encoded by the position and color of boxes.

5.3 Visualization of Chronological Data

The ThemeRiver visualization shows the thematic changes of a collection of documents as a set of "rivers" along a time line from left to right [5]. Each river represents a theme and the strength of the theme at a certain point in time is depicted by the width of the river.

Other visualization of events along a time axis [8, 11] focus on the duration of events or when an event has been sent or received. In particular, for the visualization of parallel systems [9] visualization with parallel time axes have been used early on. In Timeline Trees we also use parallel time axes but with a focus on the concurrent participation in transactions.

Furthermore, we have to mention that the notion of Timeline Trees has already been indroduced in [6] but is used for a different concept: There, Timeline Trees describe branching timelines in an interactive multimedia scenario.

6. CONCLUSIONS

We have introduced Timeline Trees as a visualization technique to explore sequences of transactions in information hierarchies. We discussed the various features of the visualization technique, and illustrated their usefulness by applying it to data sets from very different domains. These applications illustrate that Timeline Trees aid users to detect

- global and local trends with respect to the frequency and strength (measure) of the transactions, and
- relations between lower and higher levels of abstraction in the information hierarchy.

7. ACKNOWLEDGMENTS

The authors wish to thank Peter Weißgerber and Felix Bott for providing data sets and constructive comments.

8. REFERENCES

[1] K. Andrews and H. Heidegger. Information slices: Visualising and exploring large hierarchies using cascading, semi-circular discs (late breaking hot topic paper). In *Proceedings of the IEEE Symposium on Information Visualization (INFOVIS'98)*, pages 9–12, Research Triangle Park, NC, 1998.

[2] M. Burch and S. Diehl. Trees in a treemap. In *Proceedings of 13th Conference on Visualization and Data Analysis (VDA 2006)*, San Jose, California, 2006.

[3] J.-D. Fekete, D. Wand, N. Dang, A. Aris, and C. Plaisant. Overlaying graph links on treemaps. In *Poster Compendium of the IEEE Symposium on Information Visualization (INFOVIS'03), Los Alamitos, CA, USA*. IEEE, 2003.

[4] P. R. Halmos. *Measure Theory*. Van Nostrand, New York, NY, 1950.

[5] S. Havre, E. Hetzler, P. Whitney, and L. Nowell. Themeriver: visualizing thematic changes in large document collections. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):9–20, 2002.

[6] N. Hirzalla, B. Falchuk, and A. Karmouch. A temporal model for interactive multimedia scenarios. *IEEE MultiMedia*, 2(3):24–31, 1995.

[7] B. Johnson and B. Shneiderman. Tree-maps: A space-filling approach to the visualization of hierarchical information structures. In *Proceedings of IEEE Visualization Conference*, pages 284–291, San Diego, CA, 1991.

[8] G. M. Karam. Visualization using timelines. In *Proceedings of the 1994 ACM SIGSOFT international Symposium on Software Testing and Analysis ISSTA'94*, pages 125–137, New York, NY, USA, 1994. ACM.

[9] E. Kraemer and J. T. Stasko. The visualization of parallel systems: An overview. *Journal of Parallel and Distributed Computing*, 18(6):36–46, 1993.

[10] P. Neumann, S. Schlechtweg, and S. Carpendale. Arctrees: Visualizing relations in hierarchical data. In K. W. Brodlie, D. J. Duke, and K. I. Joy, editors, *Data Visualization 2005, Eurographics/IEEE VGTC Symposium on Visualization Symposium Proceedings*, pages 53–60, Aire-la-Ville, Switzerland, 2005. The Eurographics Association.

[11] C. Plaisant, B. Milash, A. Rose, S. Widoff, and B. Shneiderman. Lifelines: visualizing personal histories. In *Proceedings of the SIGCHI conference on Human factors in computing systems CHI'96*, New York, NY, USA, 1996. ACM.

[12] E. M. Reingold and J. S. Tilford. Tidier drawing of trees. *IEEE Transactions on Software Engineering*, 7(2):223–228, 1981.

[13] Slava Pestov and Contributors. JEDIT project homepage. `http://www.jedit.org/`.

[14] J. Stasko and E. Zhang. Focus+context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations. In *Proceedings of the Symposium on Information Visualization (InfoVis'00), Salt Lake City, UT*, pages 57–65, Washington, DC, 2000. IEEE Computer Society Press.

[15] J. J. van Wijk and H. van de Wetering. Cushion treemaps: Visualization of hierarchical information. In *Proceedings of IEEE Symposium on Information Visualization INFOVIS'99*, pages 73–78, San Francisco, 1999.

[16] K.-P. Yee, D. Fisher, R. Dhamija, and M. Hearst. Animated exploration of dynamic graphs with radial layout. In *Proceedings of the IEEE Symposium on Information Visualization (INFOVIS'01), San Diego, CA, USA*. IEEE, 2001.

[17] S. Zhao, M. J. McGuffin, and M. H. Chignell. Elastic hierarchies: Combining treemaps and node-link diagrams. In *Proceedings of the IEEE Symposium on Information Visualization (INFOVIS'05), Minneapolis, MN, USA*. IEEE, 2005.