Evaluating the Readability of Extended Filter/Flow Graphs

Florian Haag* Steffen Lohmann† Thomas Ertl‡

Institute for Visualization and Interactive Systems
University of Stuttgart, Germany

Abstract

The filter/flow model is a graph-based query visualization capable of representing arbitrary Boolean expressions. However, the resulting graphs quickly become large and hard to handle when representing complex search queries. We developed an extended filter/flow model that allows the display of complex queries in a more compact form. This paper reports on a user study we conducted to evaluate the readability of the extended model. The results indicate that it is as readable as the basic one and slightly preferred by users. Therefore, we regard the extended model as a good alternative to the basic one, especially when it comes to the visualization of complex queries.

Index Terms: H.5.2 [Information Interfaces and Presentation]: User Interfaces—Graphical user interfaces (GUI)

1 Introduction

The filter/flow model [15] is a popular metaphor and graphical representation that helps users understand search queries. Boolean expressions are represented as directed acyclic graphs. The edges stand for the flow of data items, while the nodes represent filter functions to block certain items. Conjunctions and disjunctions are expressed by nodes connected sequentially and in parallel, negations by inverted colors. These simple but powerful principles allow for the representation of arbitrary Boolean expressions.

An exemplary filter/flow graph for querying a database of hotels is depicted in Figure 1. It filters hotels and determines their suitability for summer or winter holidays. For example, the creator of this graph wants to select hotels for winter holidays that are (among other criteria) below 800 meters above sea level, or that are between 800 and 1500 meters above sea level and have a sauna. Note that, in filter/flow terminology, the whole graph forms a *query*, with 'Winter Holidays' as one *result set*. The number of intermediary results is sometimes represented by the thickness of the edges [15]. This is not depicted in the figure, as it is not relevant in this work.

A known issue of filter/flow graphs is their limited scalability: They quickly become large and hard to handle when representing more complex search queries (as hinted at in Figure 1). Complex search queries may occur in various fields, amongst others in extensive medical databases [7] or in tourism-related information stores [9]. Even though the graphs may still be readable, they can contain a large number of nodes and edges occupying display space and complicating manual editing in interactive implementations.

To overcome this limitation, we developed an extended filter/flow model that equips nodes with multiple emitters [3]. Each emitter has different semantics and an individual set of flows. This enables the definition of filter nodes that express several conditions at a time, which in turn results in more compact graphs.

*e-mail: florian.haag@vis.uni-stuttgart.de
†e-mail: steffen.lohmann@vis.uni-stuttgart.de
‡e-mail: thomas.ertl@vis.uni-stuttgart.de

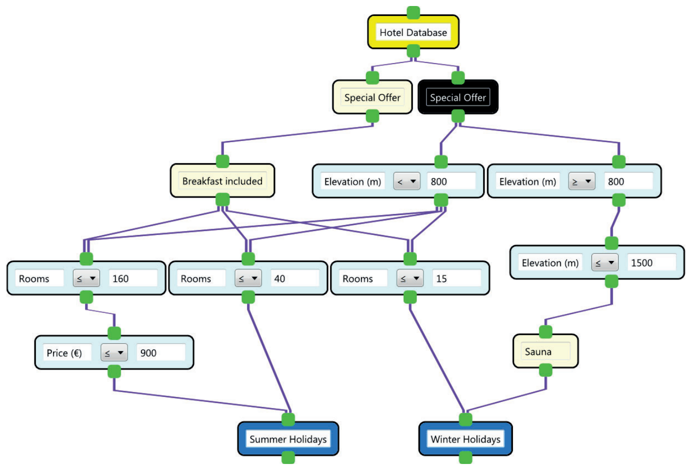


Figure 1: A basic filter/flow graph, depicting a query to a hotel database.

Figure 2(a) shows an extended filter/flow graph representing the same Boolean expression as the basic one of Figure 1. Several of the nodes feature more than one emitter and incorporate filter functions that previously had to be expressed by multiple nodes. As a result, the overall clarity of the extended graph has improved: It contains fewer nodes and edges (9 nodes instead of 14, 11 edges instead of 18), and has a lower number of duplicated and inverted filter nodes.

The basic idea of the extended filter/flow model is related to that of Yahoo! Pipes [12], though with important differences: While Yahoo! Pipes focuses on rearranging, sorting and transforming data records, our focus remains with the original idea of using filter/flow graphs for visual querying. Any basic filter/flow graph can be transformed into the extended model (and vice versa). In [3], we have formally described that transformation process and defined some specialized filter nodes for recurring subgraph patterns. Examples of these filter nodes, such as *binary operator*, *conditional*, and *between* filters, are included in Figure 2.

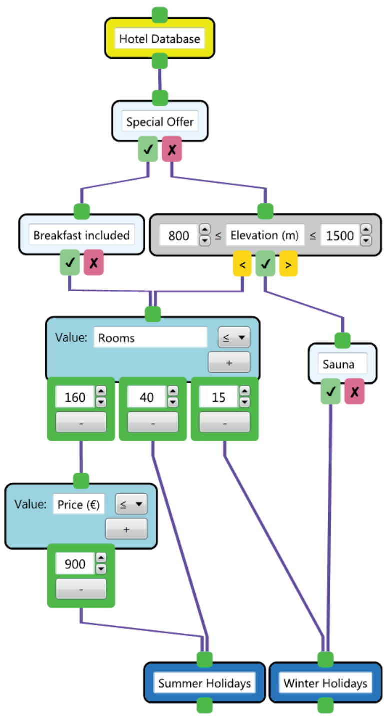
We expect the extended filter/flow model to be of value for the field, as it provides a compact alternative to basic filter/flow graphs. It can save a lot of display space and reduce visual clutter. This is particularly useful in the representation of complex search queries that include various sets of alternative or mutually exclusive filters.

However, these advantages are only useful if the extended model is at least of similar perceived and actual readability as the basic one. In order to verify this, we conducted a user study to compare the readability of extended filter/flow graphs with that of basic ones. Before we report and discuss the study in Sections 3 and 4, we provide some basic considerations and related work in Section 2.

2 Basic Considerations and Related Work

Several advancements to the original filter/flow model have been proposed in the past. Common examples include the coloring of nodes, a user-chosen graph layout, or the use of multiple result sets [2, 4]. These modifications can be considered general improve-

This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in "Proceedings of the 2013 Graphics Interface Conference", http://dl.acm.org/citation.cfm?id=2532129.2532136



(a)

Name: Central Hotel
Breakfast included: Yes
Elevation: 1981m
Price: 1425€
Rooms: 38
Sauna: Yes
Special Offer: Yes

Result: {Summer Holidays}

Name: Grand Hotel
Breakfast included: Yes
Elevation: 702m
Price: 479€
Rooms: 214
Sauna: No
Special Offer: No

Result: $\emptyset$

Name: Valley Inn
Breakfast included: No
Elevation: 240m
Price: 516€
Rooms: 8
Sauna: Yes
Special Offer: No

Result: {Winter Holidays, Summer Holidays}

(b)

Figure 2: (a) An extended filter/flow graph for the same query as Figure 1, along with (b) three exemplary data records and their results when filtered by the graph.

ments to the visual representation of the original filter/flow model rather than a conceptual change. Hence, we also incorporated them into both the basic and the extended graphs used in this work.

As we are particularly interested in complex queries, we do not expect users to instantly understand the complete set of conditions for a particular result set in a filter/flow graph. The only viable method to grasp the meaning of such a graph is by tentatively studying its effect on exemplary data items. Therefore, we are interested in learning if users are able to look at the graph and find out whether a given data item could reach one or more result sets or not.

Similar approaches are used in related studies that evaluate the readability of query visualizations, including the one in the original work on the filter/flow model conducted by Young and Shneiderman [15]: Here, participants were provided with a set of employee records and a query graph, for which they had to determine which records would be part of the result set. In Facet-Streams [6], the readability was evaluated by asking participants whether given records could reach indicated locations in the graph or not. The readability of queries built in the visual trace modeling language (VTML) [8] was compared to that of SQL queries by asking users to select the correct answers for given queries. Similarly, users had to find the correct interpretations for ready-made visualizations in the recent evaluation of a time-based filtering technique [1].

Regarding the thematic context of the user tasks, we decided for simple scenarios that can easily be explained to study participants, such as finding suitable cars or apartments. Even though we expect extended filter/flow graphs to be particularly useful in complex domains, such as technical or scientifical databases, we wanted to avoid any bias that may result from a misunderstanding of the scenario. This is common practice in the evaluation of visual querying approaches, as also reflected by related studies that use similar thematic contexts, such as finding hotels [6], real estate offers [4], people in an employee database [11], or digital cameras [2].

All filter/flow graphs used in the study were created with an editor we have developed for this purpose. The editor is a standalone application implemented in C# for the Microsoft .NET Framework. It allows for the insertion and unrestricted placement of filter nodes that can be connected by automatically positioned flows. Editing of nodes takes place directly in the graph, which is why the graph depictions—actually screenshots of the editor—include user interface elements. This is a realistic trait, because filter/flow graphs are usually part of visual querying tools. Therefore, user interface elements are frequently visible in the visual representations [14, 13]. As the distinction between the basic and the extended model is based on different filter node types, our editor supports both models by offering two alternative sets of nodes. The two example graphs shown in this paper have also been created with this editor. In the following, we report on the study in detail.

3 User Study

The main objective of the user study was to evaluate the readability of extended filter/flow graphs by comparing them to basic ones. Since we developed the extended model to overcome limitations of large filter/flow graphs, we were particularly interested in complex search queries represented by many nodes and edges. As noted above, our hypothesis was that the extended filter/flow graphs can be read at least as well as basic filter/flow graphs.

3.1 Participants

The study was conducted with 28 participants (21 male, 7 female) between 19 and 30 years of age ($M = 24.5$ years) who were recruited from the technical university campus. All of the participants were studying or working in a technology or science oriented discipline, such as aerospace engineering, environmental engineering, architecture, computer science, maths, or physics. We considered this selection beneficial as it would allow us to observe how people from a variety of fields read filter/flow graphs. All of these disciplines may come in contact with large amounts of information of different kinds, and people working in these disciplines may be required to create complex search queries.

22 of the participants stated to have at least a basic understanding of programming concepts. Three more had some previous knowledge about Boolean logic or electrical circuit diagrams, which have some conceptual similarities to filter/flow visualizations. As nowadays virtually all technology-oriented majors include some basic computer science education, this prevalence of knowledge related to basic programming concepts including Boolean logic was expected and considered representative for the target group.

3.2 Materials

We devised six queries to imaginary databases for the study. Each query included between two and four result sets. For each of these queries, we used the aforementioned editor to design two filter/flow graphs, one based on the basic model and one on the extended model. This ensured that there were six basic and six extended filter/flow graphs representing queries of the same complexity. Each participant was shown the basic filter/flow graph for some of these queries and the extended one for the others, but never both the basic and the extended graph for the same query. In addition, we prepared a brief scenario description for each of the queries.

The graphs looked like the ones depicted in Figure 1 and Figure 2(a), but contained more nodes and edges than these examples. The basic filter/flow graphs used in the study had between 32 and 53 nodes ($M = 39.7$) and between 41 and 100 edges ($M = 66$); the corresponding extended graphs had 16 to 29 nodes ($M = 21.2$) and 30 to 44 edges ($M = 34.7$). Again, this difference in graph size was due to the more compact representation in the extended filter/flow model, while query complexity remained constant.[1] A 22” TFT monitor with a screen resolution of 1920 × 1200 pixels was used in full-screen mode to display the graphs.

For each query, we created ten data records by assigning a variety of values to the attributes the query was composed of. Three exemplary data records for the graphs shown in this paper are depicted in Figure 2(b). We printed these records onto paper cards.

3.3 Design

Each participant was shown both basic and extended filter/flow graphs. That way we were able to ask for a comparison of the two models. Participants were assigned to one of two groups in order to prevent carryover effects that may result from remembering queries. Members of each group were shown three queries as basic filter/flow graphs that the other group saw as extended graphs and vice versa for the remaining three queries. Hence, no participant saw both the basic and the extended graph of the same query.

In addition, we counterbalanced the order in which the participants viewed the two models. One subgroup of participants started with the basic filter/flow graphs, while the other group first viewed the extended ones. For each participant, the study comprised of 30 trials, with each of the trials using one of the aforementioned data record cards. The order of queries and data record cards was randomized, as was the selection of the cards.

In sum, we used a 2×4 mixed design with type of model (basic vs. extended) as independent variable tested within subjects. The four conditions resulting from counterbalancing the order of the models and the query sets were tested between subjects (cp. Table 1). Participants were randomly assigned to one of the four conditions. The dependent variables were time and accuracy of task completion as well as user preference for one of the models.

Table 1: Study design: 4 conditions (C), each with 7 participants (N) and 30 trials (15 per model type), counter-balancing the two model types (M1 & M2) and two query sets (Q1 & Q2).

			within subjects (randomized)	
	C	**N**	**15 trials***	**15 trials***
between subjects (randomized)	1	7	M1 / Q1	M2 / Q2
	2	7	M1 / Q2	M2 / Q1
	3	7	M2 / Q1	M1 / Q2
	4	7	M2 / Q2	M1 / Q1

*3 graphs in randomized order, each presented with 5 randomly selected data records.

3.4 Procedure

Participants took part in the study one at a time. They first received a printed introduction to the general filter/flow concept. We illustrated this introduction with a ‘neutral’ filter/flow graph consisting of only simple filter nodes (i.e. no negated, duplicated, or specialized filter nodes). Participants were instructed to ask any questions needed to fully understand the concept. Four introductory examples were provided to let participants get used to the experimental setup, using another neutral filter/flow graph displayed on screen.

For every participant, the study was divided into two parts, each using either the basic or the extended filter/flow model. At the beginning of every such part, participants were given a description of the specifics of the graph model used in that part (for example, the way negations or conditions are visualized). Then, participants had to pass five trials for each of the three graphs. In every trial, they were given a data record card. After getting an overview of its attributes and values for a moment, a graph was shown on screen. Participants had to find out which result sets in the graph could be reached by the current data record. Answers were noted and the time was automatically stored based on a key press by the participants after answering the question. A blank screen was shown at the end of each trial to prevent memorization of the graph and to allow a break if needed. The first two trials were for practice, the other three were scored.

After completion of all trials, participants had to compare the basic and extended filter/flow model in a questionnaire. They had to state their preference for one of the models and could write down perceived advantages and disadvantages of each model. The remaining questions asked for demographic data about the participants. Both types of graphs were neutrally labeled as ‘Graph type A’ and ‘Graph type B’ throughout the whole user study. Overall, the study took about 50 to 60 minutes per participants, for which the participants were financially compensated.

Answers were considered correct if all result sets and only the result sets reached by the selected data record were stated. To illustrate this, Figure 2(b) includes the expected answers for the three exemplary data records. As the number of result sets in the graphs varied between two and four, the scores of all questions were normalized. In other words, in a graph with n result sets, correctly stating whether a particular result set was reachable would add $\frac{1}{n}$ to the score for the question.

As mentioned, the duration between the participants’ first glance at the graph and their completion of the answer was measured. This measurement was performed with the custom tool that displayed the graphs, hence the measured timespan would exactly match the time during which graphs were visible on screen. Participants were instructed to focus primarily on accuracy so as to avoid guessing.

3.5 Results

We used a mixed ANOVA to compare the scores obtained with the basic and extended filter/flow model and test for interactions between the four conditions. Table 2 shows means and standard deviations (SD) for the time and accuracy values. The values were similar in all four conditions and did not seem to be influenced by order effects ($F(3,24) = .287, p = .834$ for accuracy, $F(3,24) = .403, p = .752$ for time).

Table 2: Average time and accuracy values; participants’ votes for each model.

Filter/flow model	**N**	**Accurracy**		**Time (in sec)**		**Votes**
		Mean	**SD**	**Mean**	**SD**	
Basic	28	.987	.027	30.1	6.7	9
Extended	28	.967	.059	26.0	6.6	19

Participants performed very well in both graph models, i.e. they made few mistakes in general. Due to the high complexity of the presented graphs, we have not anticipated such good scores.

The score differences between the two models were small and non-significant ($F(1,24) = 2.63, p = .118$). The differences in task completion time, by contrast, were significant ($F(1,24) = 19.25, p < .001$). On average, participants completed the tasks four seconds (or 14%) faster with the extended filter/flow graph model compared to the basic one.

Furthermore, the extended filter/flow model was slightly preferred by the participants, as two thirds voted for this model in the questionnaire. The following attributes were given as reasons for the choice by more than four participants: ‘more clear’ (16×), ‘faster’ (8×), and ‘more compact’ (6×). Accordingly, the basic filter/flow model received some negative comments that can be sum-

[1]The tested graphs are available as supplemental material to this paper.

marized with 'too many edges' (8×) and 'visual clutter' (5×). Critical comments concerning the extended model were mainly related to its visual appearance. Four participants found that the specialized filter nodes partly contained too much information on a small space, and two more stated that especially the relational operator can be easily overlooked in some cases. However, overall the participants quickly understood how to interpret the extended filter/flow graphs and reported no serious difficulties in reading them.

4 Discussion and Conclusion

We have compared the readability of basic and extended filter/flow graphs. Our goal was to evaluate if users can read the extended model as flawlessly and quickly as the basic one. Our hypothesis was that both models are equally readable; in other words, that the improved scalability of the extended model does not go along with a decrease in readability.

The results of the study support this hypothesis. Despite their more compact representation and specialized filter nodes, the extended filter/flow graphs seem to be as readable as the basic ones. The results even indicate that they could be read slightly faster than basic ones. These findings are also supported by the votes and comments of the study participants, since a majority of them preferred the extended filter/flow model.

As an additional result of the study, we found that participants managed to use filter/flow graphs that incorporated several result sets. We had expected this, as participants would have to evaluate intermediate results after every filter node anyway, thereby also following various paths in the graph at a time. While multiple result sets were already successfully used in works such as FindFlow [4] or LARK [14], it is interesting to know that they are also usable in more complex graphs. This opens up promising possibilities for complex queries with several similar result sets that can be compared to each other, and for collaborative filtering scenarios where users jointly work on queries by defining group restrictions and combine those with personal restrictions to derive various customized result sets.

Visual interfaces for complex querying are an increasingly important topic in expert systems and visual analytics environments that often have to deal with large sets of multivariate data [5, 2, 1]. We hope that our findings pave the way for the visual representation of more complex queries than the ones usually presented along with filter/flow-based approaches [4, 6, 11]. Related work suggests handling complex graphs by defining reusable subqueries [4, 10]. A combination of that feature with the extended filter/flow model may further increase the scalability of the graphs. The extended filter/flow model could also be combined with other filter/flow-based enhancements, for example work focusing on the adaptation of the concept for special tasks and environments [6, 14].

The tasks tested in the study were relatively straightforward, as users just had to follow edges in the graph. On the other hand, it is this aspect that makes the concept simple and allowed our study participants to understand the idea behind the filter/flow model with only little training. This held true even though we had not incorporated any direction markers, such as arrows, in the graph depictions; in our case, a simple convention that flows always enter at one side of filter nodes and exit at the other was sufficient to understand the graphs.

As mentioned above, all graphs displayed during the study had the look and feel imposed by our editor application, including the aforementioned general improvements such as coloring of nodes. However, we used the same look and feel for the depictions of both the basic and extended filter/flow graphs. The editor does therefore not provide any additional benefits to the extended model that would not be present in the basic one. For that reason, we consider the danger that this created any bias towards the extended graph model to be minimal.

Finally, it should be noted that we evaluated only the readability in this work, while aspects such as the users' ability to construct or extend filter/flow graphs were not tested. We are planning to do so in future work by providing scenario-specific filter nodes and evaluate the extended graph model in a variety of use cases. Overall, we hope that other filter/flow-based approaches for visual querying as well as graph-based visualization techniques will benefit from the insights gained in this user study.

Acknowledgements

We would like to thank all participants of the user study. We are also grateful to our colleagues Florian Alt and Niels Henze for their advice on the design and evaluation of the study, as well as our student research assistant Hao Li for her support in conducting the user study. This research was funded through the IP-KOM-ÖV project (German Ministry of Economy and Technology (BMWi) grant number 19P10003N).

References

[1] C. Combi and B. Oliboni. Visually defining and querying consistent multi-granular clinical temporal abstractions. *Artif. Intell. Med.*, 54(2):75–101, 2012.

[2] N. Elmqvist, J. Stasko, and P. Tsigas. DataMeadow: A visual canvas for analysis of large-scale multivariate data. *Information Visualization*, 7(1):18–33, 2008.

[3] F. Haag, S. Lohmann, and T. Ertl. Simplifying filter/flow graphs by subgraph substitution. In *Proc. of the 2012 IEEE Symposium on Visual Languages and Human-Centric Computing*, VL/HCC '12, pages 145–148. IEEE, 2012.

[4] T. Hansaki, B. Shizuki, K. Misue, and J. Tanaka. FindFlow: Visual interface for information search based on intermediate results. In *Proc. of the 2006 Asia-Pacific Symposium on Information Visualisation*, APVis '06, pages 147–152. ACS, 2006.

[5] T. Huan, A. Y. Sivachenko, S. H. Harrison, and J. Y. Chen. ProteoLens: A visual analytic tool for multi-scale database-driven biological network data mining. *BMC Bioinformatics*, 9 (Suppl 9)(S5), 2008.

[6] H.-C. Jetter, J. Gerken, M. Zöllner, H. Reiterer, and N. Milic-Frayling. Materializing the query with facet-streams: a hybrid surface for collaborative search on tabletops. In *Proc. of the 2011 Annual Conference on Human Factors in Computing Systems*, CHI '11, pages 3013–3022. ACM, 2011.

[7] D. Klimov, Y. Shahar, and M. Taieb-Maimon. Intelligent selection and retrieval of multiple time-oriented records. *J. Intell. Inf. Syst.*, 35(2):261–300, 2010.

[8] P. Mäder and J. Cleland-Huang. A visual language for modeling and executing traceability queries. *Software & Systems Modeling*, 2012. online first, doi: 10.1007/s10270-012-0237-0.

[9] H. Meuss, K. U. Schulz, and F. Bry. Visual querying and exploration of large answers in xml databases with X^2: A demonstration. In *Proc. of the 19th International Conference on Data Engineering*, ICDE '03, pages 777–779. IEEE, 2003.

[10] A. Morris, A. Abdelmoty, B. El-Geresy, and C. Jones. A filter flow visual querying language and interface for spatial databases. *GeoInformatica*, 8(2):107–141, 2004.

[11] N. Murray, N. Paton, and C. Goble. Kaleidoquery: A visual query language for object databases. In *Proc. of the 4th International Working Conference on Advanced Visual Interfaces*, AVI '98, pages 247–257. ACM, 1998.

[12] P. Sadri, E. Ho, J. Trevor, K. Cheng, and D. Raffel. Yahoo! Pipes. http://pipes.yahoo.com/, February 2007.

[13] I. Seifert. A pool of queries: Interactive multidimensional query visualization for information seeking in digital libraries. *Information Visualization*, 10(2):97–106, 2011.

[14] M. Tobiasz, P. Isenberg, and S. Carpendale. Lark: Coordinating co-located collaboration with information visualization. *IEEE Trans. Vis. Comput. Graph.*, 15(6):1065 –1072, 2009.

[15] D. Young and B. Shneiderman. A graphical filter/flow representation of boolean queries: a prototype implementation and evaluation. *J. Am. Soc. Inf. Sci.*, 44(6):327–339, 1993.