

検索過程を確認しながら条件指定が行える検索インタフェース

Visual Search Interface for Constructing Queries while Confirming Results

飯崎 智之*1
Tomoyuki Hansaki

志築 文太郎*2
Buntarou Shizuki

三末 和男*2
Kazuo Misue

田中 二郎*2
Jiro Tanaka

*1 筑波大学大学院理工学研究科

Master's Program in Science and Engineering, University of Tsukuba

*2 筑波大学大学院システム情報工学研究科

Graduate School of Systems and Information Engineering, University of Tsukuba

In this paper, we describe the new search interface called "FindFlow." FindFlow is the search interface that user can construct queries visually on the screen. Because the constructed queries show the process of the search, user can take a step forward the search while confirming the process. We can re-combine the queries freely. It is also possible to reuse the previous search results. In addition, FindFlow shows results interactively for each operation. Therefore, we can utilize FindFlow to the search in which the user repeats trial and error many times.

1. はじめに

今日では様々な場面で検索が行われているが、目的となる情報がなかなか見つからない場合がよくある。このときユーザは、条件作成、検索実行、結果確認といった試行を何度も繰り返して適切な条件を探そうとする。また、その過程では以前の検索結果を参照することも考える。

ところが、既存の検索インタフェースは、検索式を全て評価した結果しか示さない。このため、どの条件がどの程度有効に働いているのか明確でなく、どの条件を修正して検索すべきかユーザが的確に把握することが難しい。また、何度か検索を行っても最後に行われた検索結果しか保持しない場合が多い。したがって、検索結果の再利用も困難であることが多い。

このような検索の問題に対し、視覚的に条件を作成できるようにしてユーザの負担を軽減しようという試みが行われ、画面上で条件を構成し検索を行う視覚的なインタフェースが提案・試作されている。地図情報の検索において条件を組み立てるGVQC [Diansheng 03]、データベースに対する検索条件を作成するFilter Flow [Degi 93]などが挙げられる。しかし、これらは、条件を全て評価した最終的な結果しか表示しない。このため、試行錯誤を何度も繰り返すような検索では従来の検索と同じような手間がかかってしまう。

そこで、我々は各条件ごとに検索結果を提示し、それらを確認しながら条件が変更できるインタラクティブな検索インタフェース FindFlow を作成した [飯崎 05]。FindFlow では、さまざまな検索条件を視覚的に構成することが可能で、スムーズで効率のよい検索が行える。

2. 設計方針

FindFlow の想定する利用状況として2つの状況を考えている。ひとつは、多数のメタデータを利用した複雑な検索を行う場合、もうひとつは適切な条件が分からず試行錯誤を何度も繰り返して検索を進めなければならない場合である。このような状況下でユーザがスムーズな検索を行えるよう、状況の把握が

容易にでき、条件の作成や変更がいつでも、簡単に行えるということに主眼を置いて設計を行った。また、ウェブやメール、新聞記事など様々な検索対象に対応できるように構成する。

2.1 インタフェース

複雑な条件が用いられる検索でも条件の構成を容易にするため、データフロー図をベースとした視覚的なインタフェースを提供する。データフロー図には、ひとつにコンポーネント間の構成を行うことが容易であること、そして、もうひとつにデータの処理される過程がひとつの図として示されるため、ユーザが理解しやすいというメリットがある [Steven 03]。このほか、検索の過程が図で示されているため、過去にさかのぼった検索条件の修正や、以前行った検索結果の再利用など、過去の検索に対する操作が簡単に行える利点が考えられる。

さらに、検索条件による影響をユーザが把握しやすくするため、アニメーションを用いて条件が波及する様子を示す。これによって、ユーザは設定した条件が検索にどのように影響するのかを視覚的に捉えることができる。

2.2 条件構成と結果表示

条件の構成は、エッジとノード、およびフィルタという3つの要素によって行う。

各条件の結果をそれぞれ確認できるように、ノードには検索結果の表示の役割を持たせる。ユーザは、ノードの表示を通して検索の状況を知ることができる。また、ユーザが任意の検索結果を検索母体にして、後の検索に簡単に利用できるように、ノードで分岐や結合を行うとそのノードで持つ検索結果のコピーや統合ができるようにする。

また、エッジは、データの受け渡しの他に条件による絞込み効果を持たせ、条件による絞込み状況が一目で分かるように太さを変えて示す。これによって、ユーザが画面を通して検索の状況を把握するのが容易になる。

なお、エッジで設定できる条件は、フィルタとしてエッジとは独立した形で用意して、ユーザが条件の組み換えを簡単に行えるようにする。

3. FindFlow

FindFlow は、前節の設計方針に基づいた、条件の追加や削除、複数条件の組み換えが画面上の簡単な操作で行える検索イン

連絡先: 飯崎智之, 〒 305-8573 茨城県つくば市天王台 1-1-1 筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻田中研究室, TEL/FAX 029-853-5165, hansaki@iplab.cs.tsukuba.ac.jp

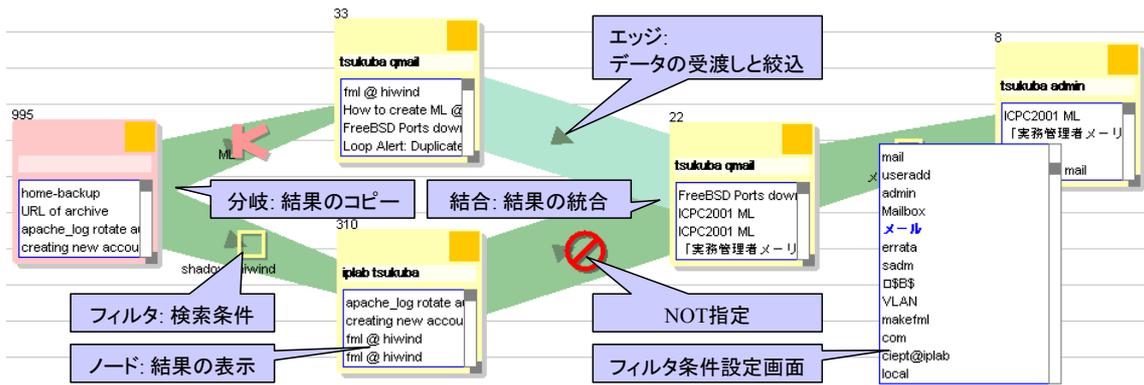


図 1: FindFlow の各要素の働き

タフェースである。検索結果は、検索式を全て評価した後の最終的な結果だけでなく、それぞれの条件に対する検索結果についても表示を行う。そのため、ユーザはどの条件が有効に機能しているかが分かる。また、ユーザの操作に対してインタラクティブに結果が反映されるため、ユーザは結果を確認しながらスムーズに条件を作成・変更できる。

3.1 構成要素

FindFlow の構成要素として、データを保持しその内容を表示するノード、ノード間の接続とデータの絞込みを行うエッジ、エッジの条件を設定するフィルタがある (図 1)。

ノード ノードはエッジからデータを受け取り、そのデータの表示を行う。つまり、そのノードに接続されているエッジで絞り込まれた結果が示されるので、ユーザはその条件が適切であるかが分かる。条件が不適切であれば、条件を変更したり、ノードやエッジを組み替えれば良い。また、表示される結果として、保持するデータと頻出キーワードを示し、ユーザがどんなデータが残っているのかを把握しやすいようにした。保持データはリストとして時系列にソートして表示し、頻出キーワードはデータ内に頻出する部分文字列の出現数をカウントしてその上位数個を表示する。

エッジ エッジはノード同士を接続し、あるノードが持つデータを別のノードへと渡す。エッジにより渡されるデータはエッジに設定されるフィルタの条件により絞り込まれる。エッジ e の始点および終点をそれぞれ、 $t(e), h(e)$ とし、エッジ上のフィルタ f によるデータ x の判定を $f(x)$ とする。構成要素 i の保持するデータ集合を $S(i)$ とすれば、エッジ e の出力するデータ集合 $S(e)$ は、

$$S(e) = \{x \in S(t(e)) | f(x)\}$$

となる。また、データがどの程度絞り込まれているかはノード間のエッジの太さで示される。集合 S の要素数を $|S|$ としたとき、エッジ e を通過する始点、終点のデータの要素数を $|S(t(e))|, |S(e)|$ 、太さを $W_t(e), W_h(e)$ 、エッジの基準の太さを C とする。 $M = \max(|S(t(e))|, |S(e)|)$ とすると、エッジ e の始点、終点の太さ $W_t(e), W_h(e)$ は、

$$W_t(e) = |S(t(e))| \div M \times C$$

$$W_h(e) = |S(e)| \div M \times C$$

となる。この太さは、データ数を比較して大きなデータ集合を持つ端を 1 とした相対的な太さで示される。そのため、データの絞

り込まれる具合が一目で分かるようになっている。

フィルタ FindFlow ではエッジをクリックしフィルタを作成することで、そのエッジでの条件が設定できる。画面上でフィルタはアイコンとして示され、ユーザはそのアイコンをエッジ間を自由に移動させて条件を設定できる。キーワードや作成者、特徴語句、作成日時などをはじめとするフィルタが作成できる。

また、フィルタのアイコンをドラッグすることで、いつでも条件の取り付け、取り外し、取り換えを行うことが可能である。フィルタのアイコンをドラッグしながら、エッジ上にポインタをあわせるとインタラクティブにそのフィルタを適用した結果を反映する。したがって、ユーザは条件組み換えをスムーズに行うことが出来る。また、フィルタのアイコンをクリックすると、条件の設定画面が表示され、フィルタ条件を詳細に決めることが出来る。条件を変更するとそれに応じて画面上でインタラクティブに反映される。

3.2 要素間の接続

ノード間の接続を行うには、入力側のノードから出力側のノードへドラッグすればよい。エッジが作成されてそのノード間を接続する。出力側のノードを指定しない場合は新しいノードが作成されて、そのノードと接続される。

また、ノードはデータの入出力に複数のエッジをとることが出来る。入力側に複数のエッジがある場合、それぞれのエッジからの入力データに対し、ユーザの指定によって、和集合もしくはは差集合を得ることができる。これは、ユーザが各エッジに対して NOT 指定することにより行われる。

ユーザの指定する集合演算には、マージもしくは差分をとることができる。ノード V に出力している n 個のエッジ e_1, e_2, \dots, e_n のうち、NOT 指定を行ったエッジ e_i, e_{i+1}, \dots, e_n があるとすると、このとき、ノード V が受け取るデータ $S(V)$ は、

$$S_{or} = \{S(e_1) \vee S(e_2) \vee \dots \vee S(e_{i-1})\}$$

$$S_{not} = \{S(e_i) \vee S(e_{i+1}) \vee \dots \vee S(e_n)\}$$

$$S(V) = S_{or} - S_{not}$$

となる。NOT 指定は、フィルタの一種として提供され、ユーザはほかの条件と同様の操作で指定することができる。

また、ノードの出力側に複数のエッジがある場合、ノードの保持するデータと同じものがそれぞれのエッジに送られる。このため、利用したい結果を持つノードから分岐させるだけで任意の時点の結果を再利用することが簡単に行える。

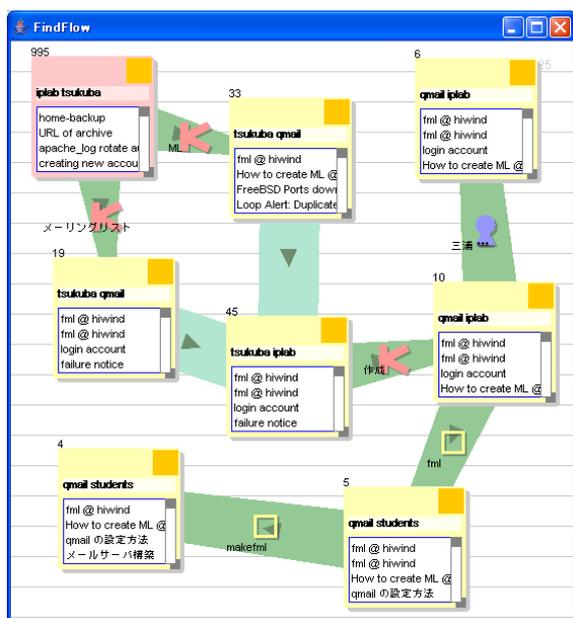


図 2: FindFlow での検索画面

3.3 要素の削除

検索条件を修正する際、ノードやエッジ、フィルタといった条件の構成要素を削除したい場合がある。FindFlow で要素の削除を行うためには、その要素にポインタを合わせて右クリックするだけでよい。その要素は削除され、反映した結果が表示される。

3.4 アニメーション

FindFlow は、データが条件により絞り込まれながら流れていく様子をアニメーションによって表現する。ユーザが検索条件を作成したり変更したりすると、その条件に応じてエッジの太さやノードの結果が検索過程の上流から下流へと変化していく。このアニメーションにより、ユーザは設定した条件が一連の検索過程にどのように影響するのかを視覚的かつ効果的に知ることができる。

3.5 条件の保存

FindFlow は条件の保存が可能であるため、各ノードは一種のデータ分類フォルダと見なして、データの分類にも利用可能である。ノードはデータの実体を保持するのではなく、検索条件に合致するデータのみを指し示す形で保持している。検索条件を変更すれば、すぐにその条件に合致したデータへと更新される。そのため、データの分類が行き詰った場合でも、条件をいちから再構成する必要がなく、必要最小限の条件変更で済む。

4. FindFlow の利用例

FindFlow は、検索対象データの読み取り部とフィルタについてはモジュール化されており、取替え可能な構成になっている。そのため、この 2 つの部分を変更するだけで様々な検索に対応できるようになっている。

具体的な例として、ここではメールを対象とした検索システムを構成した。メールの検索に対応するフィルタとして、件名、送信者、受信日時、メーリングリスト (X-ML-Name ヘッダの条件)、使用メーラー (X-Mailer ヘッダの条件) を用意した。

図 2 はメーリングリストを作成する手順が書かれたメールをアーカイブから検索した画面である。この検索の流れとして、

はじめにキーワード「ML」「メーリングリスト」指定してそれらの結果を統合した。その次に、何名かの送信者を推測して条件を作成し、さらに、頻出語句「リスト作成」を指定した。しかし、検索がうまく進まなくなってしまった。そのため、キーワード条件を新たに作成して、既存の条件と取り替えるなどの条件変更を繰り返して、最終的に「作成」という条件を設定した。その後、さらに頻出語句「fml」「makefml」という語を条件として設定した。これで、目的の情報が載っているメール「How to create ML @ fml」が見つかり、条件の作成が終了した。結果として、目的のメールを得るための条件は、「ML」または「メーリングリスト」を含み、なおかつ「作成」「fml」「makefml」を含む」であった。

このように、適切な条件を見つけるために何度も条件を変更して検索を行っていくという試行錯誤が必要となった。FindFlow を適用したシステムでは、条件の組み換えが自由に行え、操作に対してインタラクティブに結果を表示するので、ユーザはそれを確認しながら適切な設定が行え、スムーズに検索を進めることができた。

また、条件を保存することで、メール検索システムをメールの分類ツールとして利用できる。もし、メールが増えるなどして分類条件を変更しなければならない場合でも、手間なく変更することが可能であり、分類の状態を適切な状態で維持できる。

5. おわりに

我々は、各条件ごとに検索結果を提示し、それらを確認しながら条件が変更できるインタラクティブな検索インタフェース FindFlow を作成した。FindFlow では、視覚的に検索条件を容易に構成でき、検索の再試行に相当する条件の組み替えや変更を行って、徐々に目的のデータを絞り込んでいくことができる。また、FindFlow を適用したシステムとして、メール検索システムを紹介した。さらに、ウェブをはじめ、画像、新聞記事、特許などの検索、加えて、ノードをひとつの分類として見立てたデータ分類ツールとしての利用法も考えられる。今後、実際に適用したシステムを作成し、評価実験を行って本インタフェースの有効性を確認したい。

参考文献

- [Steven 03] Steven L. Tanimoto: Programming in a Data Factory, Proceedings of Human Centric Computing Language and Environments, pp.100 - 108, 2003.
- [Diansheng 03] Diansheng Guo: A Geographic Visual Query Composer (GVQC) for Accessing Federal Databases, Proceedings of National Conference for Digital Government Research, pp. 397-400, 2003.
- [Degi 93] Degi Young, Ben Shneiderman: A Graphical Filter Flow Representation of Boolean Queries: A Prototype Implementation and Evaluation, Journal of the American Society for Information Science 44(6), 327-339, 1993.
- [飯崎 05] 飯崎智之, 佐藤大介, 志築文太郎, 三末和男, 田中二郎, 検索過程を確認しながら条件指定が行える検索インタフェース, インタラクシオン 2005 論文集, pp. 189-190, 2005.