Adaptable Filter Graphs

Towards Highly-Configurable Query Visualizations

Florian Haag
Florian.Haag@vis.uni-stuttgart.de

September 20, 2012
Problem Description
Problem Description
Problem Description
Problem Description

A

December

B
Problem Description
Problem Description
Problem Description
Problem Description
Problem Description

A

December
30
min
40
min

B

30 min

40 min

www.vis.uni-stuttgart.de

Florian Haag

Adaptable Filter Graphs
Problem Description
Problem Description

A

December
30
min

B

December
40
min

30
min

40
min

www.vis.uni-stuttgart.de

Florian Haag

Adaptable Filter Graphs

2012-09-20
Problem Description
Problem Description

A → B

A

December

30 min

B

December

40 min

C

PM

C

www.vis.uni-stuttgart.de

Florian Haag

Adaptable Filter Graphs

University of Stuttgart

Germany
Problem Description
Problem Description

[Diagram showing connections between A, B, and C with times and dates.]
Problem Description
Problem Description
Problem Description

Conventional Solution

![Image of a website interface](image-url)
Problem Description

Conventional Solution
Problem Description

Conventional Solution
Problem Description

- Query visualizations
- Many query visualizations are available
  - require transformed query expressions (CNF, DNF)
  - very powerful, yet complicated to learn (InfoCrystal)
  - focus on visualizing all possible combinations of terms (Venn diagrams, InfoCrystal)
The Filter/Flow Concept

- Expression is modeled as a directed graph
- Nodes are filters (atomic terms)
- Edges are data flows
  - Number of intermediate results can be mapped to edge thickness
Our Query as a Filter/Flow Graph

∃connection c1 → c1.start = A → c1.dest = B → c1.departure = Dec 1

∃train t1 in c1.trains → t1.ridetime ≥ 30min → t1.hasRestaurantCar

∃stop s1 in c1.stops → s1.duration ≥ 40min → t1.hasRestaurant

∃connection c2 → c2.start = B → c2.dest = A → c2.departure = Dec 6

∃stop s2 in c2.stops → s2.duration ≥ 2h → s2.arrival < 6 PM → s2.place = C

s3.place ≠ C → s3.arrival ≥ 6 PM → ∀stop s3 in c2.stops

∀stop s3 in c2.stops → s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.arrival < 6 PM

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM

¬s3.place = C

¬s3.arrival < 6 PM

¬s3.place = C

¬s3.arrival ≥ 6 PM
Our Query as a Filter/Flow Graph

∃connection c1
  c1.start = A
  c1.dest = B
  c1.departure = Dec 1

∃train t1 in c1.trains
  t1.ridetime ≥ 30min
  t1.hasRestaurantCar

∃stop s1 in c1.stops
  s1.duration ≥ 40min
  t1.hasRestaurant

∃connection c2
  c2.start = B
  c2.dest = A
  c2.departure = Dec 6

∃stop s2 in c2.stops
  s2.arrival < 6 PM
  s2.place = C
  s2.duration ≥ 2h
  s3.place ≠ C

∀stop s3 in c2.stops
  s3.arrival ≥ 6 PM
  s3.duration < 2h

special discount = true
special discount = false

A
B

Florian Haag
Adaptable Filter Graphs
Problem Description

Our Query as a Filter/Flow Graph

∃connection c1
  c1.start = A
  c1.dest = B
  c1.departure = Dec 1

∃train t1 in c1.trains
  t1.ridetime ≥ 30 min
  t1.hasRestaurant

∃stop s1 in c1.stops
  s1.duration ≥ 40 min
  s1.place = C

∃connection c2
  c2.start = B
  c2.dest = A
  c2.departure = Dec 6

∃stop s2 in c2.stops
  s2.duration ≥ 2h
  s2.arrival < 6 PM
  s2.place = C

∀stop s3 in c2.stops
  s3.place ≠ C
  s3.arrival ≥ 6 PM

∃connection c1
  c1.duration ≥ 2h
  special discount = true

∃connection c1
  c1.duration < 2h
  special discount = false

∃connection c2
  c2.duration ≥ 2h

∃connection c2
  c2.duration < 2h
Problem Description

Our Query as a Filter/Flow Graph

∃connection \( c_1 \)
\[ \begin{align*}
& c_1.\text{start} = A \\
& c_1.\text{dest} = B \\
& c_1.\text{departure} = \text{Dec 1}
\end{align*} \]

∃train \( t_1 \) in \( c_1.\text{trains} \)
\[ \begin{align*}
& t_1.\text{ridetime} \geq 30 \text{min} \\
& t_1.\text{hasRestaurantCar}
\end{align*} \]

∃stop \( s_1 \) in \( c_1.\text{stops} \)
\[ \begin{align*}
& s_1.\text{duration} \geq 40 \text{min} \\
& t_1.\text{hasRestaurant}
\end{align*} \]

∃connection \( c_2 \)
\[ \begin{align*}
& c_2.\text{start} = B \\
& c_2.\text{dest} = A \\
& c_2.\text{departure} = \text{Dec 6}
\end{align*} \]

∃stop \( s_2 \) in \( c_2.\text{stops} \)
\[ \begin{align*}
& s_2.\text{arrival} < 6 \text{ PM} \\
& s_2.\text{place} = C \\
& s_2.\text{duration} \geq 2h
\end{align*} \]

∀stop \( s_3 \) in \( c_2.\text{stops} \)
\[ \begin{align*}
& s_3.\text{place} \neq C \\
& s_3.\text{arrival} \geq 6 \text{ PM}
\end{align*} \]
Problem Description

Our Query as a Filter/Flow Graph

∃connection c1
   c1.start = A
   c1.dest = B
   c1.departure = Dec 1

∃train t1 in c1.trains
   t1.ridetime ≥ 30min
   t1.hasRestaurantCar

∃stop s1 in c1.stops
   s1.duration ≥ 40min
   t1.hasRestaurant

∃connection c2
   c2.start = B
   c2.dest = A
   c2.departure = Dec 6

∃stop s2 in c1.stops
   s2.duration ≥ 2h
   c1.duration < 2h
   special discount = false

∃stop s3 in c2.stops
   s3.place ≠ C
   s3.arrival ≥ 6 PM

∀stop s1 in c2.stops
   s1.arrival < 6 PM

A B
December
December
30 min
40 min
A B
December
Problem Description

Our Query as a Filter/Flow Graph

∃connection c1
  c1.start = A
  c1.dest = B
  c1.departure = Dec 1

∃train t1 in c1.trains
  t1.ridetime ≥ 30min
  t1.hasRestaurantCar

∃stop s1 in c1.stops
  s1.duration ≥ 40min
  t1.hasRestaurant

∃connection c2
  c2.start = B
  c2.dest = A
  c2.departure = Dec 6

∃stop s2 in c2.stops
  s2.arrival < 6 PM
  s2.place = C
  s2.duration ≥ 2h

s3.place ≠ C
∀stop s3 in c2.stops
  s3.arrival ≥ 6 PM

∃connection c1
  c1.duration ≥ 2h
  special discount = true

∃connection c2
  c1.duration < 2h
  special discount = false

A B
December
December
AB
30min
40min
C
XII
III
VI
IX
PM
2h
Our Query as a Filter/Flow Graph

∃connection c1
  c1.start = A
  c1.dest = B
  c1.departure = Dec 1

∃train t1 in c1.trains
  t1.ridetime ≥ 30min
  t1.hasRestaurantCar

∃stop s1 in c1.stops
  s1.duration ≥ 40min
  t1.hasRestaurant

∃connection c2
  c2.start = B
  c2.dest = A
  c2.departure = Dec 6

∃stop s2 in c2.stops
  s2.arrival < 6 PM
  s2.place = C
  s2.duration ≥ 2h
  s3.place ≠ C

∀stop s3 in c2.stops
  s3.arrival ≥ 6 PM

∀stop s in c2.stops
  s.arrival ≥ 6 PM

special discount = true
  ∧ c1.duration ≥ 2h

special discount = false
  ∧ c1.duration < 2h

December

A

B

30
min

40
min

C

XII
III
VI
IX
PM

2h

2h

University of Stuttgart
Germany
Problem Description

Our Query as a Filter/Flow Graph

∃connection c1
  c1.start = A
  c1.dest = B
  c1.departure = Dec 1

∃train t1 in c1.trains
  t1.ridetime ≥ 30min
  t1.hasRestaurantCar

∃stop s1 in c1.stops
  s1.duration ≥ 40min
  t1.hasRestaurant

∃connection c2
  c2.start = B
  c2.dest = A
  c2.departure = Dec 6

∃stop s2 in c2.stops
  s2.arrival < 6 PM
  s2.place = C
  s2.duration ≥ 2h
  s3.place ≠ C

∀stop s3 in c2.stops
  s3.arrival ≥ 6 PM

∃connection c1
  c1.start = A
  c1.dest = B
  c1.departure = Dec 1

∀connection c2
  c2.start = B
  c2.dest = A
  c2.departure = Dec 6

special discount = true

special discount = false

2h

40min

30min

42€
Problem Description

Our Query as a Filter/Flow Graph

∃connection c1
  c1.start = A
  c1.dest = B
  c1.departure = Dec 1
∃train t1 in c1.trains
  t1.ridetime ≥ 30min
  t1.hasRestaurantCar
∃stop s1 in c1.stops
  s1.duration ≥ 40min
  t1.hasRestaurant
∃connection c2
  c2.start = B
  c2.dest = A
  c2.departure = Dec 6
∃stop s2 in c2.stops
  s2.arrival < 6 PM
  s2.place = C
  s2.duration ≥ 2h
  s3.place ≠ C
∀stop s3 in c2.stops
  s3.arrival ≥ 6 PM
  c1.duration ≥ 2h
special discount = true
特也 discount = false

A

B

December

A

B

42€

University of Stuttgart
Germany

Florian Haag

Adaptable Filter Graphs

2012-09-20 6 / 17
Problem Description

Our Query as a Filter/Flow Graph

\[ \exists \text{connection } c_1 \]
\[ c_1.\text{start} = A \quad c_1.\text{dest} = B \quad c_1.\text{departure} = \text{Dec 1} \]
\[ \exists \text{train } t_1 \text{ in } c_1.\text{trains} \]
\[ t_1.\text{ridetime} \geq 30\text{min} \quad t_1.\text{hasRestaurantCar} \]
\[ \exists \text{stop } s_1 \text{ in } c_1.\text{stops} \]
\[ s_1.\text{duration} \geq 40\text{min} \quad t_1.\text{hasRestaurant} \]

\[ \exists \text{connection } c_2 \]
\[ c_2.\text{start} = B \quad c_2.\text{dest} = A \quad c_2.\text{departure} = \text{Dec 6} \]
\[ \exists \text{stop } s_2 \text{ in } c_2.\text{stops} \]
\[ s_2.\text{arrival} < 6\text{ PM} \quad s_2.\text{place} = C \]
\[ s_2.\text{duration} \geq 2\text{h} \]

\[ \forall \text{stop } s_3 \text{ in } c_2.\text{stops} \]
\[ s_3.\text{place} \neq C \quad s_3.\text{arrival} \geq 6\text{ PM} \]

\[ c_1.\text{duration} \geq 2\text{h} \quad \text{special discount} = \text{true} \]
\[ c_1.\text{duration} < 2\text{h} \quad \text{special discount} = \text{false} \]
Solution

- Customize filter/flow graphs
- Different transformations for different users, devices, ...

- Substitute single nodes and subgraphs
Solution

The Transformation Schema

1. access
2. generic
3. use case
4. context
The Transformation Schema: Step 1

- Apply access restrictions
- Graph representation with access tags
  - any further transformations only in subgraphs with a uniform access level
Solution

The Transformation Schema: Step 1

∃connection c1  c1.start = A  c1.dest = B  c1.departure = Dec 1

∃train t1 in c1.trains

∃stop s1 in c1.stops

∃connection c2  c2.start = B  c2.dest = A  c2.departure = Dec 6

∃stop s2 in c2.stops

∀stop s3 in c2.stops

∀stop s3 in c2.stops

special discount = true

special discount = false

s3.arrival ≥ 6 PM
Solution

The Transformation Schema: Step 2

- Apply generic substitutions
- Graph representation with substitution options
The Transformation Schema: Step 2

for example:

\[
\exists \text{connection } c_1 \quad c_1.\text{start} = A \quad c_1.\text{dest} = B
\]
\[
\exists \text{train } t_1 \text{ in } c_1.\text{trains} \quad t_1.\text{ridetime} \geq 30\text{min} \quad t_1.\text{hasRestaurantCar}
\]
\[
\exists \text{stop } s_1 \text{ in } c_1.\text{stops} \quad s_1.\text{duration} \geq 40\text{min} \quad t_1.\text{hasRestaurant}
\]
\[
\text{special discount} = \begin{cases} 
\text{false} & \text{if } c_1.\text{duration} \geq 2\text{h} \\
\text{true} & \text{if } \text{yes} 
\end{cases}
\]
\[
\exists \text{connection } c_2 \quad c_2.\text{start} = B \quad c_2.\text{dest} = A
\]
\[
\exists \text{stop } s_2 \text{ in } c_2.\text{stops} \quad s_2.\text{place} = C \quad s_2.\text{duration} \geq 2\text{h}
\]
\[
\text{if } 
\begin{cases} 
\text{yes} & \text{if } s_2.\text{arrival } = \text{AM} \\
\text{no} & \text{if } s_2.\text{arrival } = \text{PM}
\end{cases}
\]

December

The Transformation Schema: **Step 3**

- Apply use case-specific substitutions
- Graph representation with more substitution options
for example:

Trip t1
- one-way trip
departure (c1):
- round trip
return (c2):

Route for t1
- start:
- destination:

Require a restaurant car on outward journey of t1
my snack stopover on outward journey of t1

if
- c1.duration ≥ 2h
  yes
  special discount = true
  no
  special discount = false

stop during return trip: s2

s2.arrival
- yes
  before
  AM
- no
  PM

s2.duration ≥ 2h

www.vis.uni-stuttgart.de
Florian Haag
Adaptable Filter Graphs
University of Stuttgart
Germany

Florian Haag
Adaptable Filter Graphs
University of Stuttgart
Germany
Solution

The Transformation Schema: Step 4

- Select substitutions based on context
- Final graph representation
Conclusion

Transformation schema for filter/flow graphs:

- Distributed collaborative access
- Scenario-based graph adaptation
- Context-based graph adaptation
Conclusion

Future Work

- Completion of the implementation of the concept
  - for tabletop display
  - for smartphones
- Evaluation of aspects of the concept in user studies
  - also considering other use cases
  - linked to semantically tagged data