

Analysis Process Modeling Notation For Business Intelligence (APMN4BI)

Thomas Neuböck

October 2017

Contents

1	Preliminaries for APMN4BI	1
1.1	Running Example	1
1.2	Enriched Dimensional Fact Model (eDFM)	1
2	Analysis Situations	7
2.1	Non-comparative Analysis Situations	7
2.2	Comparative Analysis Situations	7
3	Navigation Operators	17
3.1	Navigation Step	17
3.2	Operators Not Involving Comparison	17
3.2.1	Operators Changing Granularity Level	17
3.2.2	Operators Changing Dice Node	17
3.2.3	Operators Changing Slice Conditions	17
3.2.4	Operators Changing Base Measure Conditions	17
3.2.5	Operators Changing Measures	24
3.2.6	Operators Changing Filters	24
3.2.7	Operators Changing Cube Access	24
3.3	Operators Involving Comparison	24
3.3.1	Operators Introducing Comparison	24
3.3.2	Operators Changing Comparison	27
3.3.3	Operators Dropping Comparison	27
3.4	Use of Analysis Situations as Cubes	27

4	Extension to Schema Level	37
4.1	Analysis Situation Schema	37
4.2	Navigation Step Schema	37
4.3	Navigation guards	37
5	Business Intelligence (BI) Analysis Graphs	49
5.1	Definition of BI Analysis Graphs	49
5.2	BI Analysis Graph Schemas	49
5.3	Instances of Analysis Graph Schemas	49
5.4	Composite analysis situation	49
	List of Figures	49
	List of Tables	56

Chapter 1

Preliminaries for APMN4BI

1.1 Running Example

see Figure 1.1

1.2 Enriched Dimensional Fact Model (eDFM)

see Figure 1.2, 1.3, 1.4

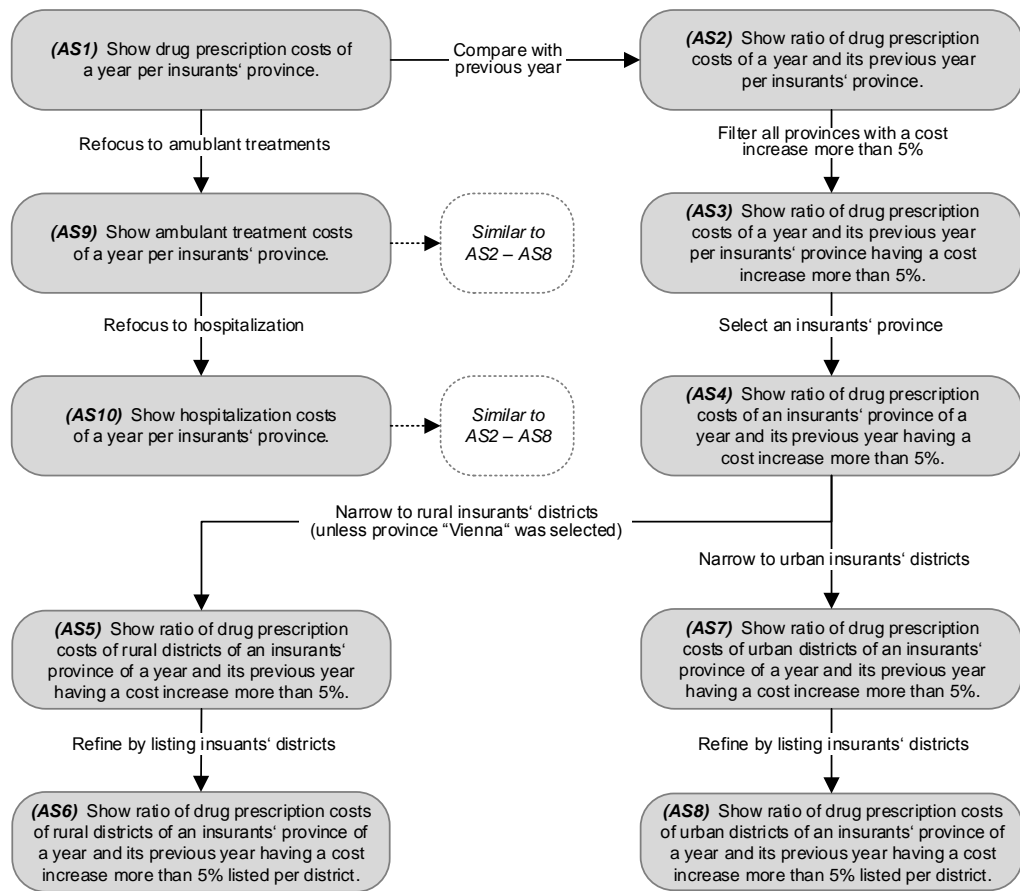


Figure 1.1: Health care use case – overview of an analysis process

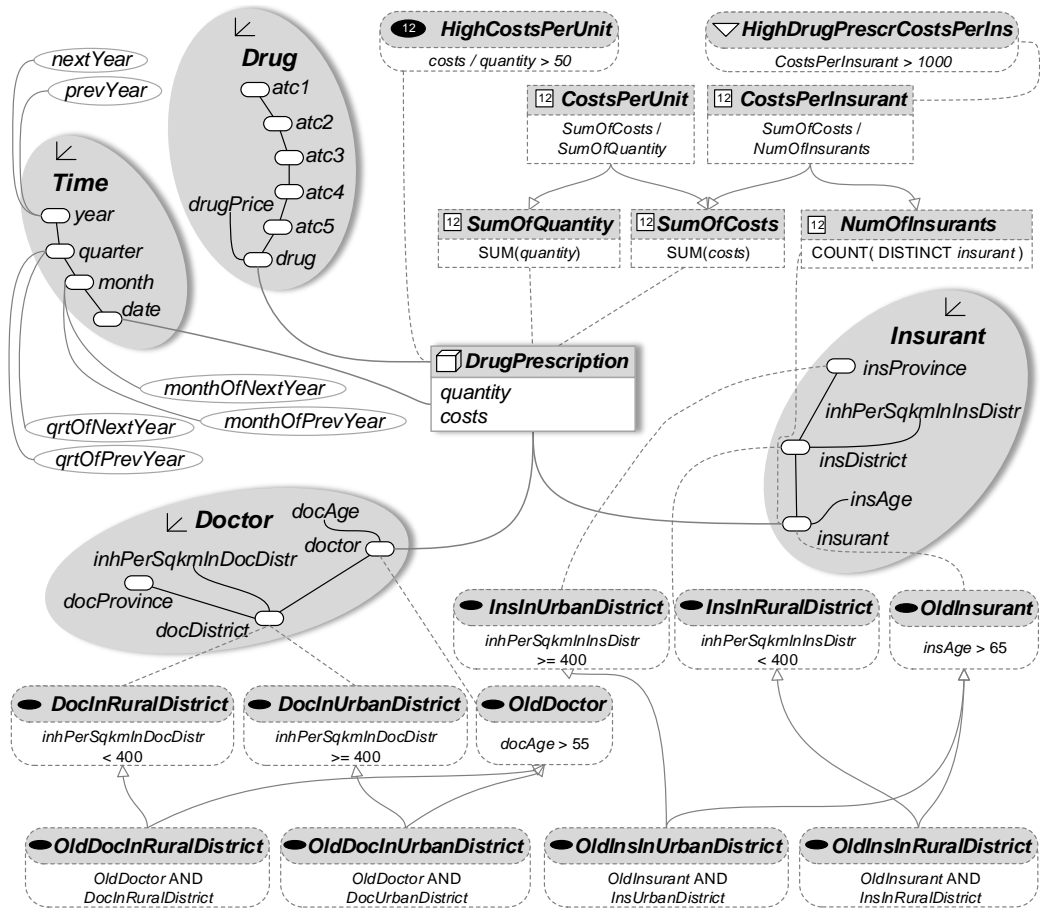


Figure 1.2: eDFM for Drug Prescriptions

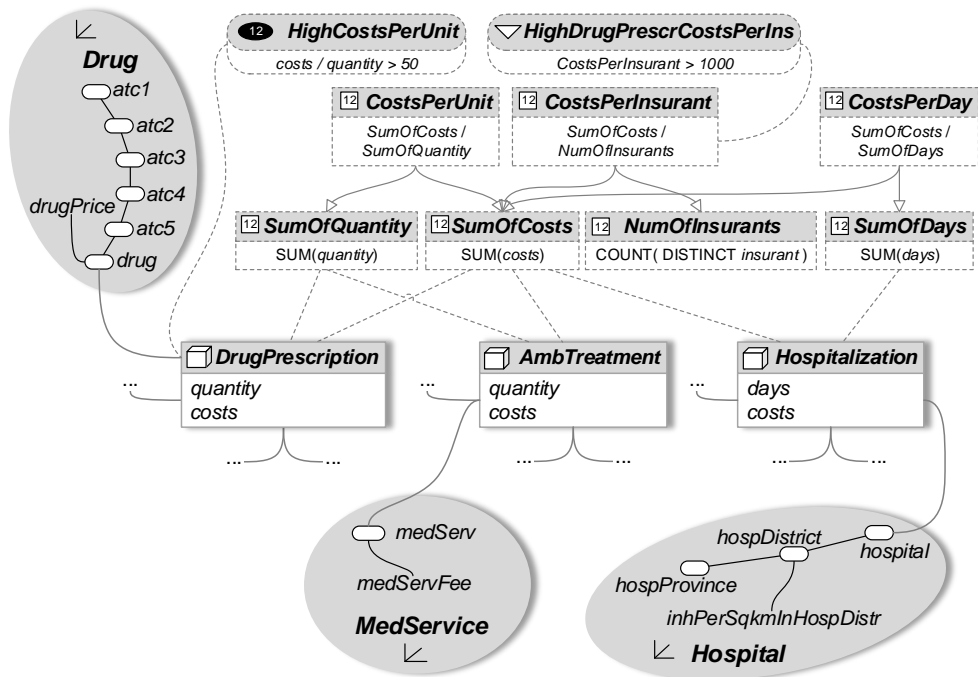


Figure 1.3: eDFM for Ambulant Treatment and Hospitalization

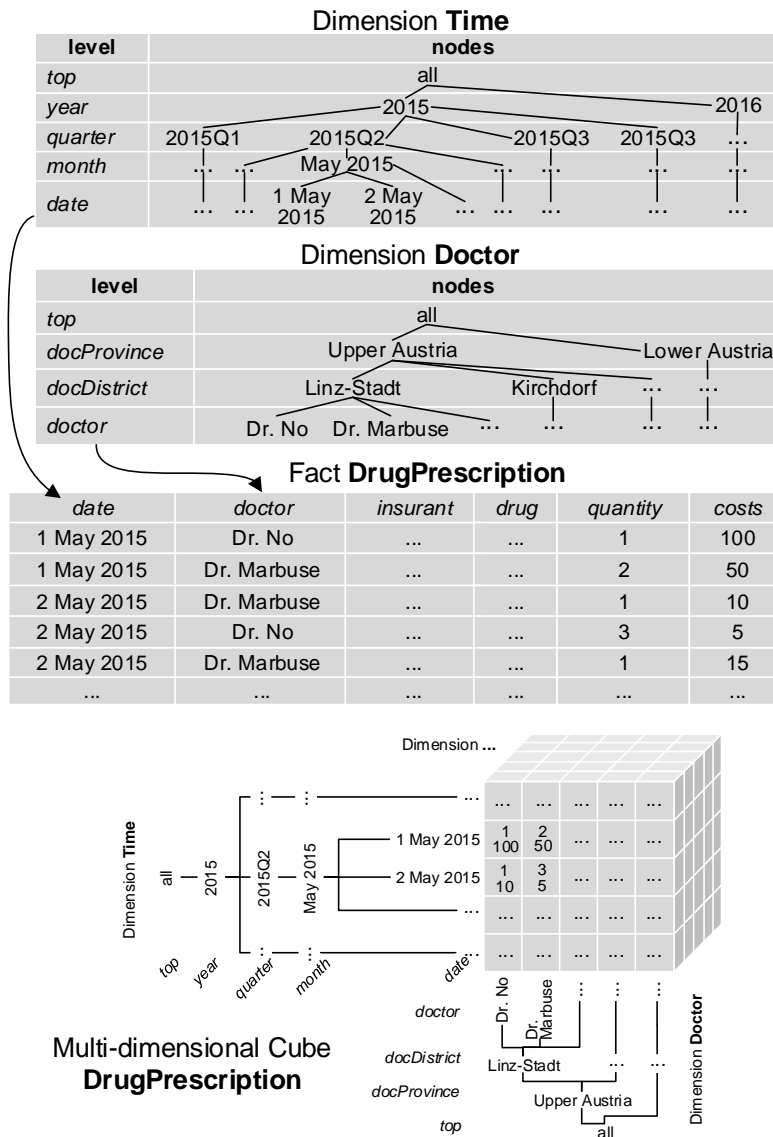


Figure 1.4: Cube Instance

Chapter 2

Analysis Situations

2.1 Non-comparative Analysis Situations

see Figure 2.1, 2.2, 2.3, 2.4, 2.5

2.2 Comparative Analysis Situations

see Figure 2.6, 2.7, 2.8, 2.9, 2.10

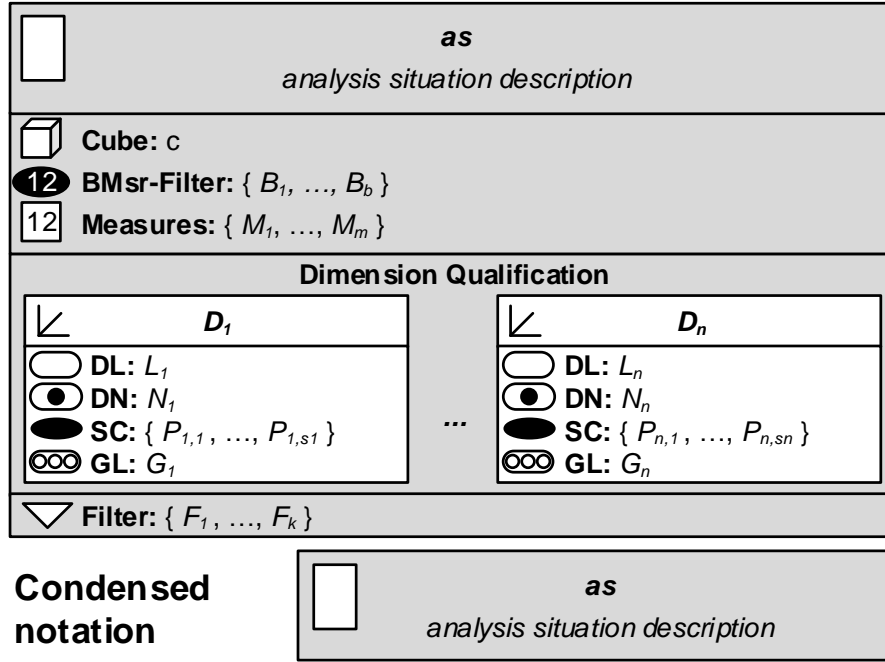


Figure 2.1: Graphical Template of a Non-comparative Analysis Situation

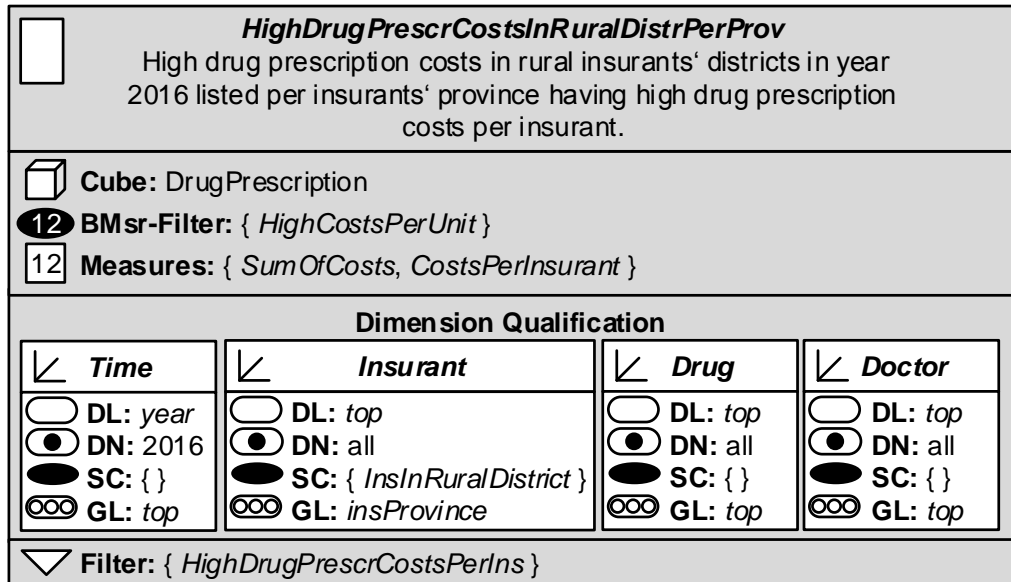


Figure 2.2: Example of a Graphical Representation of a Non-comparative Analysis Situation

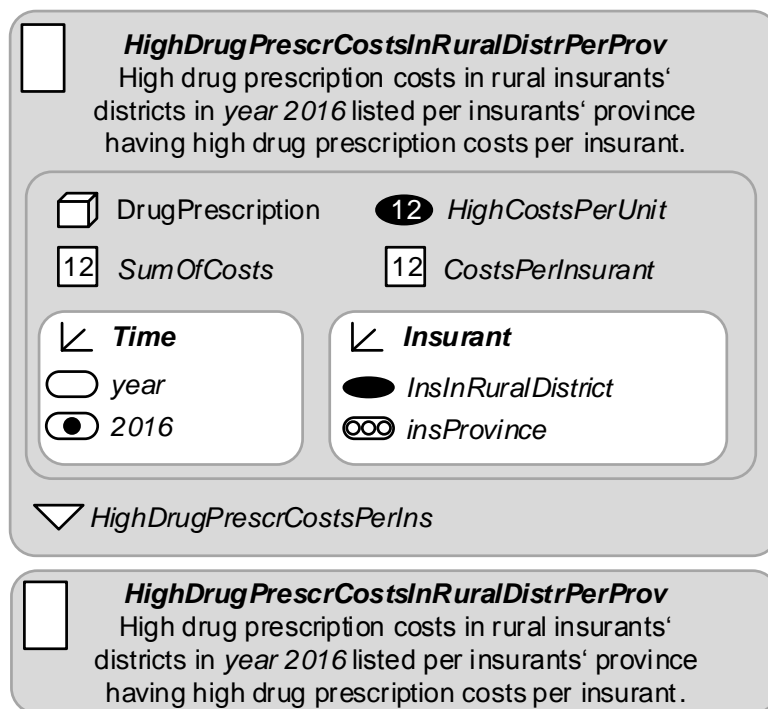


Figure 2.3: Example of an Alternative Graphical Representation of a Non-comparative Analysis Situation

SQL translation:

```

select       $G_1, \dots, G_n,$ 
             $M_1, \dots, M_m,$ 
from        c natural join  $D_1$  natural join ... natural join  $D_n$ 
where        $B_1$  and ... and  $B_b$           and
             $L_1 = N_1$  and ... and  $L_n = N_n$     and
             $P_{1,1}$  and ... and  $P_{1,s1}$         and
            ...                             and
             $P_{n,1}$  and ... and  $P_{n,sn}$ 

group by     $G_1, \dots, G_n$ 
having       $F_1$  and ... and  $F_k$ 

```

Result set:

G_1	...	G_n	M_1	...	M_m
...
...
...

Figure 2.4: SQL Translation

SQL translation:

```

select      insProvince,
            SUM(costs)                                as SumOfCosts,
            SUM(costs) / COUNT( DISTINCT insurant )    as CostsPerInsurant
from        DrugPrescription      natural join
            Time                  natural join
            Insurant
where       costs / quantity > 50 and
            year = 2015 and
            inhPerSqkmInInsDistr < 400
group by    insProvince
having      SUM(costs) / COUNT( DISTINCT insurant ) > 1000

```

Example result set:

<i>insProvince</i>	<i>SumOfCosts</i>	<i>CostsPerInsurant</i>
Upper Austria	117790612.01	2120.31
Lower Austria	197790612.45	2415.07
...

Figure 2.5: Example of an SQL Translation

<div style="display: flex; justify-content: space-between; align-items: center;"> <div> <input type="checkbox"/> : <input type="checkbox"/> </div> <div> cas <i>description of the comparative analysis situation</i> </div> </div>			
<input type="checkbox"/> Context of Interest (CoI)		<input type="checkbox"/> Context of Comparison (CoC)	
<div style="display: flex; justify-content: space-between;"> <div> <input type="checkbox"/> Cube: c <input checked="" type="checkbox"/> BMSr-Filter: $\{B_1, \dots, B_b\}$ <input type="checkbox"/> Measures: $\{M_1, \dots, M_m\}$ </div> <div> <input type="checkbox"/> Cube: c' <input checked="" type="checkbox"/> BMSr-Filter: $\{B'_{1'}, \dots, B'_{b'}\}$ <input type="checkbox"/> Measures: $\{M'_{1'}, \dots, M'_{m'}\}$ </div> </div>		<div style="display: flex; justify-content: space-between;"> <div> <input type="checkbox"/> Cube: c <input checked="" type="checkbox"/> BMSr-Filter: $\{B'_{1'}, \dots, B'_{b'}\}$ <input type="checkbox"/> Measures: $\{M'_{1'}, \dots, M'_{m'}\}$ </div> <div> <input type="checkbox"/> Cube: c' <input checked="" type="checkbox"/> BMSr-Filter: $\{B'_{1'}, \dots, B'_{b'}\}$ <input type="checkbox"/> Measures: $\{M'_{1'}, \dots, M'_{m'}\}$ </div> </div>	
Dimension Qualification		Dimension Qualification	
<div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> <input type="checkbox"/> DL: L_1 <input checked="" type="checkbox"/> DN: N_1 <input checked="" type="checkbox"/> SC: $\{P_{1,1}, \dots, P_{1,s1}\}$ <input checked="" type="checkbox"/> GL: G_1 </div> <div style="width: 10%; text-align: center;">...</div> <div style="width: 45%;"> <input type="checkbox"/> DL: L_n <input checked="" type="checkbox"/> DN: N_n <input checked="" type="checkbox"/> SC: $\{P_{n,1}, \dots, P_{n,sn}\}$ <input checked="" type="checkbox"/> GL: G_n </div> </div>	<div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> <input type="checkbox"/> DL: L'_1 <input checked="" type="checkbox"/> DN: N'_1 <input checked="" type="checkbox"/> SC: $\{P'_{1,1}, \dots, P'_{1,s'1}\}$ <input checked="" type="checkbox"/> GL: G'_1 </div> <div style="width: 10%; text-align: center;">...</div> <div style="width: 45%;"> <input type="checkbox"/> DL: $L'_{n'}$ <input checked="" type="checkbox"/> DN: $N'_{n'}$ <input checked="" type="checkbox"/> SC: $\{P'_{n',1}, \dots, P'_{n',s'n'}\}$ <input checked="" type="checkbox"/> GL: $G'_{n'}$ </div> </div>	<div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> <input type="checkbox"/> DL: L'_1 <input checked="" type="checkbox"/> DN: N'_1 <input checked="" type="checkbox"/> SC: $\{P'_{1,1}, \dots, P'_{1,s'1}\}$ <input checked="" type="checkbox"/> GL: G'_1 </div> <div style="width: 10%; text-align: center;">...</div> <div style="width: 45%;"> <input type="checkbox"/> DL: $L'_{n'}$ <input checked="" type="checkbox"/> DN: $N'_{n'}$ <input checked="" type="checkbox"/> SC: $\{P'_{n',1}, \dots, P'_{n',s'n'}\}$ <input checked="" type="checkbox"/> GL: $G'_{n'}$ </div> </div>	<div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> <input type="checkbox"/> DL: L'_1 <input checked="" type="checkbox"/> DN: N'_1 <input checked="" type="checkbox"/> SC: $\{P'_{1,1}, \dots, P'_{1,s'1}\}$ <input checked="" type="checkbox"/> GL: G'_1 </div> <div style="width: 10%; text-align: center;">...</div> <div style="width: 45%;"> <input type="checkbox"/> DL: $L'_{n'}$ <input checked="" type="checkbox"/> DN: $N'_{n'}$ <input checked="" type="checkbox"/> SC: $\{P'_{n',1}, \dots, P'_{n',s'n'}\}$ <input checked="" type="checkbox"/> GL: $G'_{n'}$ </div> </div>
<input type="checkbox"/> Filter: $\{F_1, \dots, F_k\}$		<input type="checkbox"/> Filter: $\{F'_{1'}, \dots, F'_{k'}\}$	
<input checked="" type="checkbox"/> Join Condition: $\{J_1, \dots, J_t\}$			
<input checked="" type="checkbox"/> Scores: $\{S_1, \dots, S_v\}$			
<input checked="" type="checkbox"/> Score Filter: $\{F^o_{1s}, \dots, F^o_w\}$			
<div style="display: flex; justify-content: space-between; align-items: center;"> <div> <input type="checkbox"/> : <input type="checkbox"/> </div> <div> cas <i>description of the comparative analysis situation</i> </div> </div>			

Figure 2.6: Graphical Template of a Comparative Analysis Situation

<input type="radio"/> : <input type="radio"/>		HighDrugPrescrCostsInRuralDistrictsComparedWithPrevYear Increases of high drug prescription costs in rural insurers' districts in year 2016 compared with previous year 2015 listed per insurers' province	
<input type="radio"/>		Context of Interest (CoI)	
<input type="checkbox"/>		Context of Comparison (CoC)	
<input type="checkbox"/>		Cube: DrugPrescription BMsr-Filter: { HighCostsPerUnit } Measures: { SumOfCosts, CostsPerInsurant }	
<input type="checkbox"/>		Dimension Qualification	
<input type="checkbox"/>		Dimension Qualification	
<input type="checkbox"/>		Filter: { HighDrugPrescrCostsPerIns }	
<input type="checkbox"/>		Join Condition: { CoI.insProvince = CoC.insProvince, CoI.topTime = CoC.topTime, CoI.CoI.topDrug = CoC.topDrug, CoI.topDoctor = CoC.topDoctor }	
<input type="checkbox"/>		Scores: { RatioOfSumOfCosts, RatioOfCostsPerInsurant }	
<input type="checkbox"/>		Score Filter: { IncreasedCostsPerInsurant }	
<input type="radio"/> : <input type="radio"/>		HighDrugPrescrCostsInRuralDistrictsComparedWithPrevYear Increases of high drug prescription costs in rural insurers' districts in year 2016 compared with previous year 2015 listed per insurers' province	

Figure 2.7: Example of a Graphical Representation of a Comparative Analysis Situation

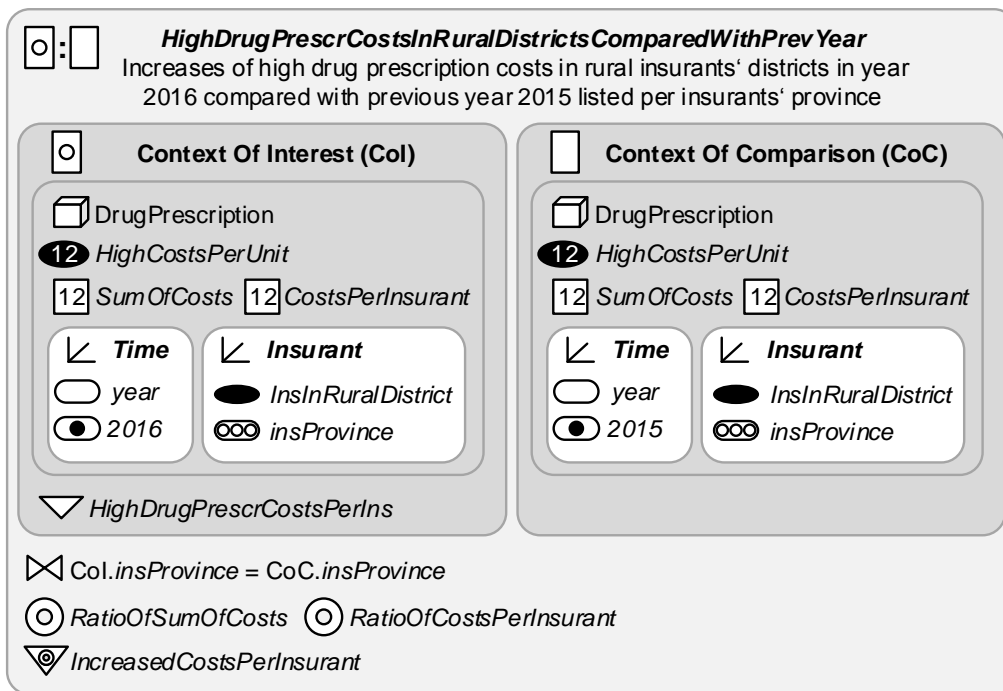


Figure 2.8: Example of an Alternative Graphical Representation of a Comparative Analysis Situation

SQL translation:

select	Col.G ₁ , ..., Col.G _n , Col.M ₁ , ..., Col.M _m , CoC.G' ₁ , ..., CoC.G' _n , CoC.M' ₁ , ..., CoC.M' _m , S ₁ , ..., S _v	
from	(select from where group by having) Col	G ₁ , ..., G _n , M ₁ , ..., M _m c natural join D ₁ natural join ... natural join D _n B ₁ and ... and B _b and L ₁ = N ₁ and ... and L _n = N _n and P _{1,1} and ... and P _{1,s1} and ... and P _{n,1} and ... and P _{n,sn} G ₁ , ..., G _n F ₁ and ... and F _k
inner join	(select from where group by having) CoC	G' ₁ , ..., G' _n , M' ₁ , ..., M' _m c' natural join D' ₁ natural join ... natural join D' _n B' ₁ and ... and B' _b and L' ₁ = N' ₁ and ... and L' _n = N' _n and P' _{1,1} and ... and P' _{1,s'1} and ... and P' _{n,1} and ... and P' _{n,s'n} G' ₁ , ..., G' _n F' ₁ and ... and F' _{k'}
on	J ₁ and ... and J _t	
where	F° ₁ and ... and F° _w	

Result set:

[illegible]

Figure 2.9: SQL Translation of a Comparative Analysis Situation

SQL translation:

```

select      Col.insProvince, Col.SumOfCosts, Col.CostsPerInsurant,
            CoC.insProvince, CoC.SumOfCosts, CoC.CostsPerInsurant,
            Col.SumOfCosts / CoC.SumOfCosts          as RatioOfSumOfCosts,
            Col.CostsPerInsurant / CoC.CostsPerInsurant as RatioOfCostsPerInsurant

from        (select      insProvince, SUM(costs) as SumOfCosts, SUM(costs) / COUNT( DISTINCT insurant ) as CostsPerInsurant
            from        DrugPrescription natural join Time natural join Insurant
            where       costs / quantity > 50 and
                        year = 2016 and
                        inhPerSqkmInInsDistr < 400
            group by    insProvince
            having       SUM(costs) / COUNT( DISTINCT insurant ) > 1000
            ) Col

inner join   (select      insProvince, SUM(costs) as SumOfCosts, SUM(costs) / COUNT( DISTINCT insurant ) as CostsPerInsurant
            from        DrugPrescription natural join Time natural join Insurant
            where       costs / quantity > 50 and
                        year = 2015 and
                        inhPerSqkmInInsDistr < 400
            group by    insProvince
            ) CoC

on          Col.insProvince = CoC.insProvince

where       Col.CostsPerInsurant / CoC.CostsPerInsurant > 1

```

Example result set:

Col. <i>insProvince</i>	Col. <i>SumOfCosts</i>	Col. <i>CostsPerInsurant</i>	CoC. <i>insProvince</i>	CoC. <i>SumOfCosts</i>	CoC. <i>CostsPerInsurant</i>	<i>RatioOfSumOfCosts</i>	<i>RatioOfCostsPerInsurant</i>
Upper Austria	117790612.01	2120.31	Upper Austria	107790612.01	1959.21	1.0928	1.0822
Lower Austria	197790612.45	2415.07	Lower Austria	187790612.45	2295.74	1.0533	1.0520
...

Figure 2.10: Example of an SQL Translation of a Comparative Analysis Situation

Chapter 3

Navigation Operators

3.1 Navigation Step

see Figure 3.1

3.2 Operators Not Involving Comparison

3.2.1 Operators Changing Granularity Level

see Table 3.1

3.2.2 Operators Changing Dice Node

see Table 3.2, 3.3

3.2.3 Operators Changing Slice Conditions

see Table 3.4

3.2.4 Operators Changing Base Measure Conditions

see Table 3.5

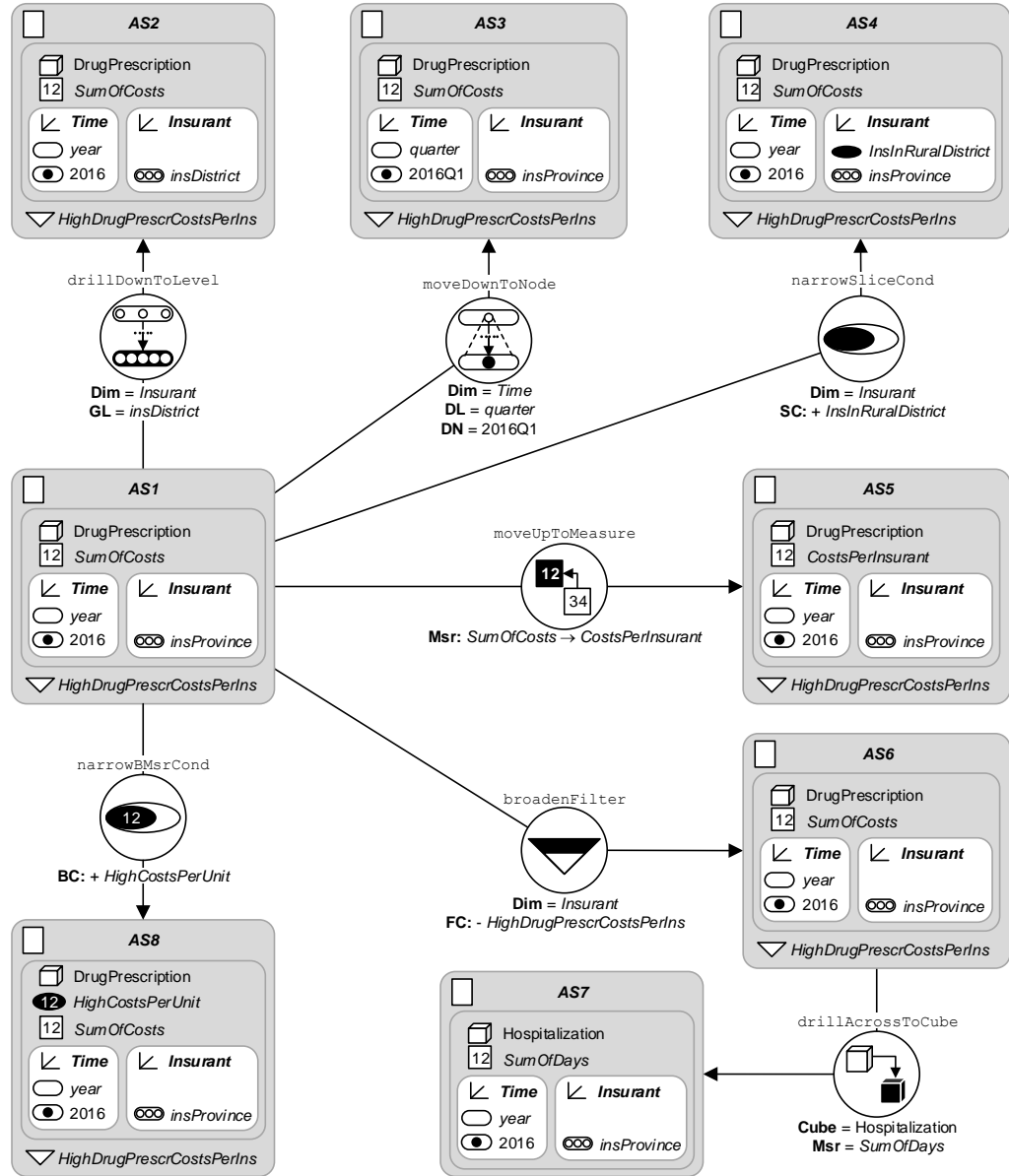


Figure 3.1: Navigation operator examples

Table 3.1: Operators `drillDownOneLevel`, `drillDownToLevel`, `rollUpOneLevel`, and `rollUpToLevel`





Operator Definition	Symbol
<u><code>drillDownOneLevel(D)</code></u> Precondition: $GranLvl_{src}(D) \neq base_D$ Postcondition: $GranLvl_{trg}(D) \rightarrow GranLvl_{src}(D)$	<code>drillDownOneLevel</code>  $Dim = D$
<u><code>drillDownToLevel(D, G)</code></u> Precondition: $G \twoheadrightarrow GranLvl_{src}(D)$ Postcondition: $GranLvl_{trg}(D) = G$	<code>drillDownToLevel</code>  $Dim = D$ $GL = G$
<u><code>rollUpOneLevel(D)</code></u> Precondition: $GranLvl_{src}(D) \neq top_D$ Postcondition: $GranLvl_{src}(D) \rightarrow GranLvl_{trg}(D)$	<code>rollUpOneLevel</code>  $Dim = D$
<u><code>rollUpToLevel(D, G)</code></u> Precondition: $GranLvl_{src}(D) \twoheadrightarrow G$ Postcondition: $GranLvl_{trg}(D) = G$	<code>rollUpToLevel</code>  $Dim = D$ $GL = G$

Table 3.2: Operators `moveDownToNode`, `moveUpToNode`, `moveAsideToNode`, and `moveToNode`





Operator Definition	Symbol
<u><code>moveDownToNode(D, L, N)</code></u> Precondition: $L \in Lvl_s_D$, $L \neq base_D$, $L \rightarrow DiceLvl_{src}(D)$, $N \in Nodes_d(L)$ with $d = DimInstance_{src}(D)$, and $N \rightarrow DiceNode_{src}(D)$ Postcondition: $L = DiceLvl_{trg}(D)$, $N = DiceNode_{trg}(D)$	moveDownToNode  Dim = D DL = L DN = N
<u><code>moveUpToNode(D, L)</code></u> Precondition: $L \in Lvl_s_D$, $L \neq top_D$, $DiceLvl_{src}(D) \rightarrow L$, $DiceNode_{src}(D) \rightarrow N$ for the unique existing $N \in Nodes_d(L)$, with $d = DimInstance_{src}(D)$ Postcondition: $L = DiceLvl_{trg}(D)$, $N = DiceNode_{trg}(D)$	moveUpToNode  Dim = D DL = L
<u><code>moveAsideToNode(D, N)</code></u> Precondition: $DiceLvl_{src}(D) \neq top_D$, $N \neq DiceNode_{src}(D)$ and, for the unique existing \hat{N} with $DiceNode_{src}(D) \rightarrow \hat{N}$: $N \rightarrow \hat{N}$ Postcondition: $DiceNode_{trg}(D) = N$	moveAsideToNode  Dim = D DN = N
<u><code>moveToNode(D, L, N)</code></u> Precondition: $L \neq DiceLvl_{src}(D)$, $L \in Lvl_s_D$, and $N \in Nodes_d(L)$, with $d = DimInstance_{src}(D)$ Postcondition: $DiceLvl_{trg}(D) = L$, $DiceNode_{trg}(D) = N$	moveToNode  Dim = D DL = L DN = N

Table 3.3: Operators `moveDownToFirstNode`, `moveDownToLastNode`, `moveToNextNode`, and `moveToPrevNode`





Operator Definition	Symbol
<u><code>moveDownToFirstNode(D)</code></u> Precondition: $DiceLvl_{src}(D) \neq base_D$ Postcondition: $DiceLvl_{trg}(D) \rightarrow DiceLvl_{src}(D)$, $DiceNode_{trg}(D) =$ $min_{\prec}\{N \mid N \rightarrow DiceNode_{src}(D)\}$	<code>moveDownToFirstNode</code>  $Dim = D$
<u><code>moveDownToLastNode(D)</code></u> Precondition: $DiceLvl_{src}(D) \neq base_D$ Postcondition: $DiceLvl_{trg}(D) \rightarrow DiceLvl_{src}(D)$, $DiceNode_{trg}(D) =$ $max_{\prec}\{N \mid N \rightarrow DiceNode_{src}(D)\}$	<code>moveDownToLastNode</code>  $Dim = D$
<u><code>moveToNextNode(D)</code></u> Precondition: $DiceLvl_{src}(D) \neq top_D$ and, for the unique existing N with $DiceNode_{src}(D) \rightarrow N$, there exists an $N' \rightarrow N$, such that $N' = next_{\prec}(DiceNode_{src}(D))$ Postcondition: $DiceNode_{trg}(D) = N'$	<code>moveToNextNode</code>  $Dim = D$
<u><code>moveToPrevNode(D)</code></u> Precondition: $DiceLvl_{src}(D) \neq top_D$ and, for the unique existing N with $DiceNode_{src}(D) \rightarrow N$, there exists an $N' \rightarrow N$, such that $N' = prev_{\prec}(DiceNode_{src}(D))$ Postcondition: $DiceNode_{trg}(D) = N'$	<code>moveToPrevNode</code>  $Dim = D$

Table 3.4: Operators narrowSliceCond, broadenSliceCond, and refocusSliceCond













Operator Definition	Symbol
$\text{narrowSliceCond}(D, +P_1, \dots, +P_s)$ Precondition: $P_1, \dots, P_s \in \text{DimPredicates}_D$, $\{P_1, \dots, P_s\} \not\subseteq \text{SliceConds}_{\text{src}}(D)$ Postcondition: $\text{SliceConds}_{\text{trg}}(D) =$ $\text{SliceConds}_{\text{src}}(D) \cup \{P_1, \dots, P_s\}$	narrowSliceCond  Dim = D SC: $+ P_1, \dots, + P_s$
$\text{narrowSliceCond}(D, P_{\text{old}} \rightarrow P_{\text{new}})$ Precondition: $P_{\text{old}}, P_{\text{new}} \in \text{DimPredicates}_D$, $P_{\text{old}} \in \text{SliceConds}_{\text{src}}(D)$, $P_{\text{new}} \notin \text{SliceConds}_{\text{src}}(D)$, $P_{\text{new}} \Rightarrow P_{\text{old}}$ Postcondition: $\text{SliceConds}_{\text{trg}}(D) =$ $\text{SliceConds}_{\text{src}}(D) \setminus \{P_{\text{old}}\} \cup \{P_{\text{new}}\}$	narrowSliceCond  Dim = D SC: $P_{\text{old}} \rightarrow P_{\text{new}}$
$\text{broadenSliceCond}(D, -P_1, \dots, -P_s)$ Precondition: $P_1, \dots, P_s \in \text{DimPredicates}_D$, $\{P_1, \dots, P_s\} \not\subseteq \text{SliceConds}_{\text{src}}(D)$ Postcondition: $\text{SliceConds}_{\text{trg}}(D) =$ $\text{SliceConds}_{\text{src}}(D) \setminus \{P_1, \dots, P_s\}$	broadenSliceCond  Dim = D SC: $- P_1, \dots, - P_s$
$\text{broadenSliceCond}(D, P_{\text{old}} \rightarrow P_{\text{new}})$ Precondition: $P_{\text{old}}, P_{\text{new}} \in \text{DimPredicates}_D$, $P_{\text{old}} \in \text{SliceConds}_{\text{src}}(D)$, $P_{\text{new}} \notin \text{SliceConds}_{\text{src}}(D)$, $P_{\text{old}} \Rightarrow P_{\text{new}}$ Postcondition: $\text{SliceConds}_{\text{trg}}(D) =$ $\text{SliceConds}_{\text{src}}(D) \setminus \{P_{\text{old}}\} \cup \{P_{\text{new}}\}$	broadenSliceCond  Dim = D SC: $P_{\text{old}} \rightarrow P_{\text{new}}$
$\text{refocusSliceCond}(D, P_1, \dots, P_s)$ Precondition: $P_1, \dots, P_s \in \text{DimPredicates}_D$, $\{P_1, \dots, P_s\} \neq \text{SliceConds}_{\text{src}}(D)$ Postcondition: $\text{SliceConds}_{\text{trg}}(D) =$ $\{P_1, \dots, P_s\}$	refocusSliceCond  Dim = D SC: P_1, \dots, P_s
$\text{refocusSliceCond}(D, P_{\text{old}} \rightarrow P_{\text{new}})$ Precondition: $P_{\text{old}}, P_{\text{new}} \in \text{DimPredicates}_D$, $P_{\text{old}} \in \text{SliceConds}_{\text{src}}(D)$, $P_{\text{new}} \notin \text{SliceConds}_{\text{src}}(D)$ Postcondition: $\text{SliceConds}_{\text{trg}}(D) =$ $\text{SliceConds}_{\text{src}}(D) \setminus \{P_{\text{old}}\} \cup \{P_{\text{new}}\}$	refocusSliceCond  Dim = D SC: $P_{\text{old}} \rightarrow P_{\text{new}}$

Table 3.5: Operators narrowBMSrCond, broadenBMSrCond, and refocusBMSrCond

Operator Definition	Symbol
<u>narrowBMSrCond($+B_1, \dots, +B_b$)</u> Precondition: $B_1, \dots, B_b \in \text{BMSrPredicates}_C$ with $C = \text{CubeSchema}_{src}$, $\{B_1, \dots, B_b\} \not\subseteq \text{BMSrConds}_{src}$ Postcondition: $\text{BMSrConds}_{trg} = \text{BMSrConds}_{src} \cup \{B_1, \dots, B_b\}$	narrowBMSrCond  BC: $+B_1, \dots, +B_b$
<u>narrowBMSrCond($B_{old} \rightarrow B_{new}$)</u> Precondition: $B_{new} \in \text{BMSrPredicates}_C$ with $C = \text{CubeSchema}_{src}$, $B_{old} \in \text{BMSrConds}_{src}$, $B_{new} \notin \text{BMSrConds}_{src}$, $B_{new} \Rightarrow B_{old}$ Postcondition: $\text{BMSrConds}_{trg} = \text{BMSrConds}_{src} \setminus \{B_{old}\} \cup \{B_{new}\}$	narrowBMSrCond  BC: $B_{old} \rightarrow B_{new}$
<u>broadenBMSrCond($-B_1, \dots, -B_b$)</u> Precondition: $\{B_1, \dots, B_b\} \subseteq \text{BMSrConds}_{src}$ Postcondition: $\text{BMSrConds}_{trg} = \text{BMSrConds}_{src} \setminus \{B_1, \dots, B_b\}$	broadenBMSrCond  BC: $-B_1, \dots, -B_b$
<u>broadenBMSrCond($B_{old} \rightarrow B_{new}$)</u> Precondition: $B_{new} \in \text{BMSrPredicates}_C$ with $C = \text{CubeSchema}_{src}$, $B_{old} \in \text{BMSrConds}_{src}$, $B_{new} \notin \text{BMSrConds}_{src}$, $B_{old} \Rightarrow B_{new}$ Postcondition: $\text{BMSrConds}_{trg} = \text{BMSrConds}_{src} \setminus \{B_{old}\} \cup \{B_{new}\}$	broadenBMSrCond  BC: $B_{old} \rightarrow B_{new}$
<u>refocusBMSrCond(B_1, \dots, B_b)</u> Precondition: $B_1, \dots, B_b \in \text{BMSrPredicates}_C$ with $C = \text{CubeSchema}_{src}$ $\text{BMSrConds}_{src} \neq \{B_1, \dots, B_b\}$ Postcondition: $\text{BMSrConds}_{trg} = \{B_1, \dots, B_b\}$	refocusBMSrCond  BC: B_1, \dots, B_b
<u>refocusBMSrCond($B_{old} \rightarrow B_{new}$)</u> Precondition: $B_{new} \in \text{BMSrPredicates}_C$ with $C = \text{CubeSchema}_{src}$, $B_{old} \in \text{BMSrConds}_{src}$, $B_{new} \notin \text{BMSrConds}_{src}$ Postcondition: $\text{BMSrConds}_{trg} = \text{BMSrConds}_{src} \setminus \{B_{old}\} \cup \{B_{new}\}$	refocusBMSrCond  BC: $B_{old} \rightarrow B_{new}$

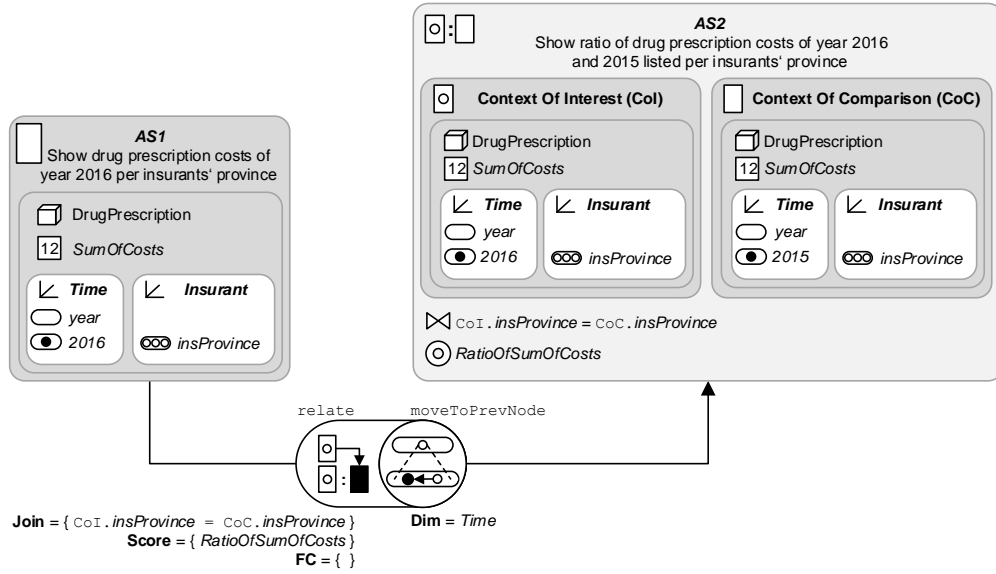


Figure 3.2: Example of a comparative navigation step

3.2.5 Operators Changing Measures

see Table 3.6

3.2.6 Operators Changing Filters

see Table 3.7

3.2.7 Operators Changing Cube Access

see Table 3.8

3.3 Operators Involving Comparison

see Figure 3.2

3.3.1 Operators Introducing Comparison

see Table 3.9

Table 3.6: Operators addMeasure, dropMeasure, refocusMeasure, moveDownToMeasure, and moveUpToMeasure







Operator Definition	Symbol
<u>addMeasure(+M₁, ..., +M_m)</u> Precondition: M ₁ , ..., M _m ∈ Msrs _C with C = CubeSchema _{src} , {M ₁ , ..., M _m } ⊄ Msrs _{src} Postcondition: Msrs _{trg} = Msrs _{src} ∪ {M ₁ , ..., M _m }	addMeasure  Msr: + M ₁ , ..., + M _m
<u>dropMeasure(-M₁, ..., -M_m)</u> Precondition: {M ₁ , ..., M _m } ⊆ Msrs _{src} Postcondition: Msrs _{trg} = Msrs _{src} \ {M ₁ , ..., M _m }	dropMeasure  Msr: - M ₁ , ..., - M _m
<u>refocusMeasure(M₁, ..., M_m)</u> Precondition: M ₁ , ..., M _m ∈ Msrs _C with C = CubeSchema _{src} , {M ₁ , ..., M _m } ≠ Msrs _{src} Postcondition: Msrs _{trg} = {M ₁ , ..., M _m }	refocusMeasure  Msr = M ₁ , ..., M _m
<u>refocusMeasure(M_{old} -> M_{new})</u> Precondition: M _{new} ∈ Msrs _C with C = CubeSchema _{src} , M _{old} ∈ Msrs _{src} , M _{new} ∉ Msrs _{src} Postcondition: Msrs _{trg} = Msrs _{src} \ {M _{old} } ∪ {M _{new} }	refocusMeasure  Msr: M _{old} → M _{new}
<u>moveDownToMeasure(M_{old} -> M_{new})</u> Precondition: M _{new} ∈ Msrs _C with C = CubeSchema _{src} , M _{new} ∉ Msrs _{src} , M _{new} → M _{old} Postcondition: Msrs _{trg} = Msrs _{src} \ {M _{old} } ∪ {M _{new} }	moveDownToMeasure  Msr: M _{old} → M _{new}
<u>moveUpToMeasure(M_{old} -> M_{new})</u> Precondition: M _{new} ∈ Msrs _C with C = CubeSchema _{src} , M _{new} ∉ Msrs _{src} , M _{old} → M _{new} Postcondition: Msrs _{trg} = Msrs _{src} \ {M _{old} } ∪ {M _{new} }	moveUpToMeasure  Msr: M _{old} → M _{new}

Table 3.7: Operator narrowFilter








Operator Definition	Symbol
narrowFilter ($+F_1, \dots, +F_k$) Precondition: $F_1, \dots, F_k \in \text{MsrPredicates}_C$ with $C = \text{CubeSchema}_{src}$, $\{F_1, \dots, F_k\} \not\subseteq \text{FilterConds}_{src}$ Postcondition: $\text{FilterConds}_{trg} = \text{FilterConds}_{src} \cup \{F_1, \dots, F_k\}$	narrowFilter  FC: $+F_1, \dots, +F_k$
narrowFilter ($F_{old} \rightarrow F_{new}$) Precondition: $F_{new} \in \text{MsrPredicates}_C$ with $C = \text{CubeSchema}_{src}$, $F_{old} \in \text{FilterConds}_{src}$, $F_{new} \notin \text{FilterConds}_{src}$, $F_{new} \Rightarrow F_{old}$ Postcondition: $\text{FilterConds}_{trg} = \text{FilterConds}_{src} \setminus \{F_{old}\} \cup \{F_{new}\}$	narrowFilter  FC: $F_{old} \rightarrow F_{new}$
broadenFilter ($-F_1, \dots, -F_k$) Precondition: $\{F_1, \dots, F_k\} \subseteq \text{FilterConds}_{src}$ Postcondition: $\text{FilterConds}_{trg} = \text{FilterConds}_{src} \setminus \{F_1, \dots, F_k\}$	broadenFilter  FC: $-F_1, \dots, -F_k$
broadenFilter ($F_{old} \rightarrow F_{new}$) Precondition: $F_{new} \in \text{MsrPredicates}_C$ with $C = \text{CubeSchema}_{src}$, $F_{old} \in \text{FilterConds}_{src}$, $F_{new} \notin \text{FilterConds}_{src}$, $F_{old} \Rightarrow F_{new}$ Postcondition: $\text{FilterConds}_{trg} = \text{FilterConds}_{src} \setminus \{F_{old}\} \cup \{F_{new}\}$	broadenFilter  FC: $F_{old} \rightarrow F_{new}$
refocusFilter (F_1, \dots, F_k) Precondition: $F_1, \dots, F_k \in \text{MsrPredicates}_C$ with $C = \text{CubeSchema}_{src}$, $\text{FilterConds}_{src} \neq \{F_1, \dots, F_k\}$ Postcondition: $\text{FilterConds}_{trg} = \{F_1, \dots, F_k\}$	refocusFilter  FC: F_1, \dots, F_k
refocusFilter ($F_{old} \rightarrow F_{new}$) Precondition: $F_{new} \in \text{MsrPredicates}_C$ with $C = \text{CubeSchema}_{src}$, $F_{old} \in \text{FilterConds}_{src}$, $F_{new} \notin \text{FilterConds}_{src}$ Postcondition: $\text{FilterConds}_{trg} = \text{FilterConds}_{src} \setminus \{F_{old}\} \cup \{F_{new}\}$	refocusFilter  FC: $F_{old} \rightarrow F_{new}$

Table 3.8: Operator `drillAcrossToCube`

Operator Definition	Symbol
<p><u><code>drillAcrossToCube(C, B, M, F)</code></u> Precondition: C is a cube, $B \subseteq BMrPredicates_C$, $M \subseteq Msrs_C$, $M \neq \emptyset$, and $F \subseteq MsrPredicates_C$ Postcondition: $CubeInstance_{trg} = C$, $BMrConds_{trg} = B$, and if $D \in DimSchemas_{src} \cap DimSchemas_C$, then $DimQual_{src}(D) \in DimQuals_{trg}$, if $D \in DimSchemas_C \wedge D \notin DimSchemas_{src}$, then $DiceLvl_{dq} = top_D$, $DiceNode_{dq} = all_D$, $SliceConds_{dq} = \emptyset$, and $GranLvl_{dq} = top_D$ with $dq = DimQual_{trg}(D)$, and moreover, $Msrs_{trg} = M$ and $FilterConds_{trg} = F$</p>	<p><code>drillAcrossToCube</code></p>  <p> Cube = C BC = B Msr = M FC = F </p>

3.3.2 Operators Changing Comparison

see Table 3.10, 3.11, 3.12, 3.13, 3.14

3.3.3 Operators Dropping Comparison

see Table 3.15

3.4 Use of Analysis Situations as Cubes

see Table 3.16

see Figure 3.3, 3.4

Table 3.9: Operators `relate` and `target`

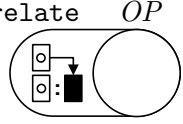
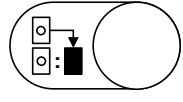
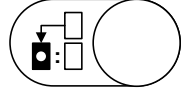
Operator Definition	Symbol
$\underline{\text{relate}(OP(p_1, \dots, p_q), J, S, F^\circ)}$ Precondition: J is a non-empty set of join conditions between src and $src.OP(p_1, \dots, p_q)$, S is a non-empty set of scores over src and $src.OP(p_1, \dots, p_q)$, and F° is a possibly empty set of score filters over src and $src.OP(p_1, \dots, p_q)$ with scores of S . Postcondition: $CoI_{trg} = src$, $CoC_{trg} = src.OP(p_1, \dots, p_q)$, $JoinConds_{trg} = J$, $Scores_{trg} = S$, $ScoreFilters_{trg} = F^\circ$	 <p> Join = J p_1 Score = S \dots FC = F° p_q </p>
$\underline{\text{relate}(J, S, F^\circ)}$ Precondition: J is a non-empty set of join conditions between src and src , S is a non-empty set of scores over src and src , and F° is a possibly empty set of score filters over src and src with scores of S . Postcondition: $CoI_{trg} = src$, $CoC_{trg} = src$, $JoinConds_{trg} = J$, $Scores_{trg} = S$, $ScoreFilters_{trg} = F^\circ$	 <p> Join = J Score = S FC = F° </p>
$\underline{\text{target}(OP(p_1, \dots, p_q), J, S, F^\circ)}$ Precondition: J is a non-empty set of join conditions between src and $src.OP(p_1, \dots, p_q)$, S is a non-empty set of scores over src and $src.OP(p_1, \dots, p_q)$, and F° is a possibly empty set of score filters over src and $src.OP(p_1, \dots, p_q)$ with scores of S . Postcondition: $CoC_{trg} = src$, $CoI_{trg} = src.OP(p_1, \dots, p_q)$, $JoinConds_{trg} = J$, $Scores_{trg} = S$, $ScoreFilters_{trg} = F^\circ$	 <p> Join = J p_1 Score = S \dots FC = F° p_q </p>

Table 3.10: Operator *rerelate*

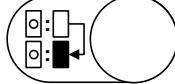
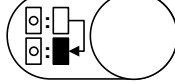
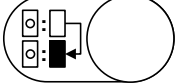
Operator Definition	Symbol
<p><u>$\text{rerelate}(OP(p_1, \dots, p_q), J)$</u> Precondition: J is a non-empty set of join conditions between CoI_{src} and $CoC_{src} \cdot OP(p_1, \dots, p_q)$ Postcondition: $CoI_{trg} = CoI_{src}$, $CoC_{trg} = CoC_{src} \cdot OP(p_1, \dots, p_q)$, $JoinConds_{trg} = J$, $Scores_{trg} = Scores_{src}$, $ScoreFilters_{trg} = ScoreFilters_{src}$</p>	<p><i>rerelate</i> OP</p>  <p>Join = J p_1 \dots p_q</p>
<p><u>$\text{rerelate}(OP(p_1, \dots, p_q), J, S, F^\circ)$</u> Precondition: J is a non-empty set of join conditions between CoI_{src} and $CoC_{src} \cdot OP(p_1, \dots, p_q)$, S is a non-empty set of scores over CoI_{src} and $CoC_{src} \cdot OP(p_1, \dots, p_q)$, and F° is a possibly empty set of score filters over CoI_{src} and $CoC_{src} \cdot OP(p_1, \dots, p_q)$ with scores of S. Postcondition: $CoI_{trg} = CoI_{src}$, $CoC_{trg} = CoC_{src} \cdot OP(p_1, \dots, p_q)$, $JoinConds_{trg} = J$, $Scores_{trg} = S$, $ScoreFilters_{trg} = F^\circ$</p>	<p><i>rerelate</i> OP</p>  <p>Join = J p_1 \dots p_q Score = S \dots FC = F° p_q</p>
<p><u>$\text{rerelate}(OP(p_1, \dots, p_q))$</u> Precondition: $JoinConds_{src}$ are also join conditions between CoI_{src} and $CoC_{src} \cdot OP(p_1, \dots, p_q)$, $Scores_{src}$ are also scores over CoI_{src} and $CoC_{src} \cdot OP(p_1, \dots, p_q)$, and $ScoreFilters_{src}$ are also scores over filters over CoI_{src} and $CoC_{src} \cdot OP(p_1, \dots, p_q)$ with scores $Scores_{src}$. Postcondition: $CoI_{trg} = CoI_{src}$, $CoC_{trg} = CoC_{src} \cdot OP(p_1, \dots, p_q)$, $JoinConds_{trg} = JoinConds_{src}$, $Scores_{trg} = Scores_{src}$, $ScoreFilters_{trg} = ScoreFilters_{src}$</p>	<p><i>rerelate</i> OP</p>  <p>p_1 \dots p_q</p>

Table 3.11: Operator retarget

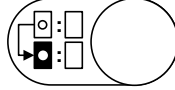
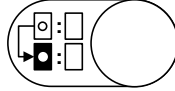
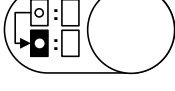
Operator Definition	Symbol
$\text{retarget}(OP(p_1, \dots, p_q), J)$ Precondition: J is a non-empty set of join conditions between $CoI_{src}.OP(p_1, \dots, p_q)$ and CoC_{src} Postcondition: $CoI_{trg} = CoI_{src}.OP(p_1, \dots, p_q)$, $CoC_{trg} = CoC_{src}$, $JoinConds_{trg} = J$, $Scores_{trg} = Scores_{src}$, $ScoreFilters_{trg} = ScoreFilters_{src}$	 <p>$Join = J$ p_1 \dots p_q</p>
$\text{retarget}(OP(p_1, \dots, p_q), J, S, F^\circ)$ Precondition: J is a non-empty set of join conditions between $CoI_{src}.OP(p_1, \dots, p_q)$ and CoC_{src} , S is a non-empty set of scores over $CoI_{src}.OP(p_1, \dots, p_q)$ and CoC_{src} , and F° is a possibly empty set of score filters over $CoI_{src}.OP(p_1, \dots, p_q)$ and CoC_{src} with scores of S . Postcondition: $CoI_{trg} = CoI_{src}.OP(p_1, \dots, p_q)$, $CoC_{trg} = CoC_{src}$, $JoinConds_{trg} = J$, $Scores_{trg} = S$, $ScoreFilters_{trg} = F^\circ$	 <p>$Join = J$ p_1 $Score = S$ \dots $FC = F^\circ$ p_q</p>
$\text{retarget}(OP(p_1, \dots, p_q))$ Precondition: $JoinConds_{src}$ are also join conditions between $CoI_{src}.OP(p_1, \dots, p_q)$ and CoC_{src} , $Scores_{src}$ are also scores over $CoI_{src}.OP(p_1, \dots, p_q)$ and CoC_{src} , and $ScoreFilters_{src}$ are also scores over filters over $CoI_{src}.OP(p_1, \dots, p_q)$ and CoC_{src} with scores $Scores_{src}$. Postcondition: $CoI_{trg} = CoI_{src}.OP(p_1, \dots, p_q)$, $CoC_{trg} = CoC_{src}$, $JoinConds_{trg} = JoinConds_{src}$, $Scores_{trg} = Scores_{src}$, $ScoreFilters_{trg} = ScoreFilters_{src}$	 <p>p_1 \dots p_q</p>

Table 3.12: Operator correlate

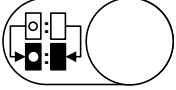
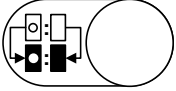
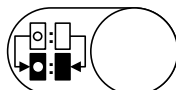
Operator Definition	Symbol
$\text{correlate}(OP(p_1, \dots, p_q), J)$ Precondition: J is a non-empty set of join conditions between $CoI_{src}.OP(p_1, \dots, p_q)$ and $CoC_{src}.OP(p_1, \dots, p_q)$ Postcondition: $CoI_{trg} = CoI_{src}.OP(p_1, \dots, p_q)$, $CoC_{trg} = CoC_{src}.OP(p_1, \dots, p_q)$, $JoinConds_{trg} = J$, $Scores_{trg} = Scores_{src}$, $ScoreFilters_{trg} = ScoreFilters_{src}$	<div style="text-align: center;"> $\text{correlate } OP$  Join = J p_1 \dots p_q </div>
$\text{correlate}(OP(p_1, \dots, p_q), J, S, F^\circ)$ Precondition: J is a non-empty set of join conditions between $CoI_{src}.OP(p_1, \dots, p_q)$ and $CoC_{src}.OP(p_1, \dots, p_q)$, S is a non-empty set of scores over $CoI_{src}.OP(p_1, \dots, p_q)$ and $CoC_{src}.OP(p_1, \dots, p_q)$, and F° is a possibly empty set of score filters over $CoI_{src}.OP(p_1, \dots, p_q)$ and $CoC_{src}.OP(p_1, \dots, p_q)$ with scores of S . Postcondition: $CoI_{trg} = CoI_{src}.OP(p_1, \dots, p_q)$, $CoC_{trg} = CoC_{src}.OP(p_1, \dots, p_q)$, $JoinConds_{trg} = J$, $Scores_{trg} = S$, $ScoreFilters_{trg} = F^\circ$	<div style="text-align: center;"> $\text{correlate } OP$  Join = J p_1 Score = S \dots FC = F° p_q </div>
$\text{correlate}(OP(p_1, \dots, p_q))$ Precondition: $JoinConds_{src}$ are also join conditions between $CoI_{src}.OP(p_1, \dots, p_q)$ and $CoC_{src}.OP(p_1, \dots, p_q)$, $Scores_{src}$ are also scores over $CoI_{src}.OP(p_1, \dots, p_q)$ and $CoC_{src}.OP(p_1, \dots, p_q)$, and $ScoreFilters_{src}$ are also scores over filters over $CoI_{src}.OP(p_1, \dots, p_q)$ and $CoC_{src}.OP(p_1, \dots, p_q)$ with scores $Scores_{src}$. Postcondition: $CoI_{trg} = CoI_{src}.OP(p_1, \dots, p_q)$, $CoC_{trg} = CoC_{src}.OP(p_1, \dots, p_q)$, $JoinConds_{trg} = JoinConds_{src}$, $Scores_{trg} = Scores_{src}$, $ScoreFilters_{trg} = ScoreFilters_{src}$	<div style="text-align: center;"> $\text{correlate } OP$  p_1 \dots p_q </div>

Table 3.13: Operators `narrowFilter`, `broadenFilter`, and `refocusFilter` for score filters



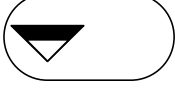

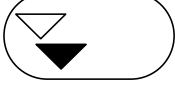
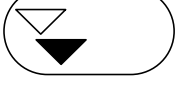
Operator Definition	Symbol
$\text{narrowFilter}(+F_1^\circ, \dots, +F_w^\circ)$ Precondition: $F_1^\circ, \dots, F_w^\circ$ are score filters over CoI_{src} and CoC_{src} with scores in $Scores_{src}$, and $\{F_1^\circ, \dots, F_w^\circ\} \not\subseteq ScoreFilters_{src}$. Postcondition: $ScoreFilters_{trg} = ScoreFilters_{src} \cup \{F_1^\circ, \dots, F_w^\circ\}$	<p>narrowFilter</p>  <p>FC: $+F_1^\circ, \dots, +F_w^\circ$</p>
$\text{narrowFilter}(F_{old}^\circ \rightarrow F_{new}^\circ)$ Precondition: F_{new}° is a score filter over CoI_{src} and CoC_{src} with scores in $Scores_{src}$, $F_{old}^\circ \in ScoreFilters_{src}$, $F_{new}^\circ \notin ScoreFilters_{src}$, and $F_{new}^\circ \Rightarrow F_{old}^\circ$. Postcondition: $ScoreFilters_{trg} = ScoreFilters_{src} \setminus \{F_{old}^\circ\} \cup \{F_{new}^\circ\}$	<p>narrowFilter</p>  <p>FC: $F_{old}^\circ \rightarrow F_{new}^\circ$</p>
$\text{broadenFilter}(-F_1^\circ, \dots, -F_w^\circ)$ Precondition: $\{F_1^\circ, \dots, F_w^\circ\} \subseteq ScoreFilters_{src}$. Postcondition: $ScoreFilters_{trg} = ScoreFilters_{src} \setminus \{F_1^\circ, \dots, F_w^\circ\}$	<p>broadenFilter</p>  <p>FC: $-F_1^\circ, \dots, -F_w^\circ$</p>
$\text{broadenFilter}(F_{old}^\circ \rightarrow F_{new}^\circ)$ Precondition: F_{new}° is a score filter over CoI_{src} and CoC_{src} with scores in $Scores_{src}$, and $F_{old}^\circ \in ScoreFilters_{src}$, $F_{new}^\circ \notin ScoreFilters_{src}$, and $F_{old}^\circ \Rightarrow F_{new}^\circ$. Postcondition: $ScoreFilters_{trg} = ScoreFilters_{src} \setminus \{F_{old}^\circ\} \cup \{F_{new}^\circ\}$	<p>broadenFilter</p>  <p>FC: $F_{old}^\circ \rightarrow F_{new}^\circ$</p>
$\text{refocusFilter}(F_1^\circ, \dots, F_w^\circ)$ Precondition: $F_1^\circ, \dots, F_w^\circ$ are score filters over CoI_{src} and CoC_{src} with scores in $Scores_{src}$, and $\{F_1^\circ, \dots, F_w^\circ\} \neq ScoreFilters_{src}$. Postcondition: $ScoreFilters_{trg} = \{F_1^\circ, \dots, F_w^\circ\}$	<p>refocusFilter</p>  <p>FC: $F_1^\circ, \dots, F_w^\circ$</p>
$\text{refocusFilter}(F_{old}^\circ \rightarrow F_{new}^\circ)$ Precondition: F_{new}° is a score filter over CoI_{src} and CoC_{src} with scores in $Scores_{src}$, $F_{old}^\circ \in ScoreFilters_{src}$, and $F_{new}^\circ \notin ScoreFilters_{src}$. Postcondition: $ScoreFilters_{trg} = ScoreFilters_{src} \setminus \{F_{old}^\circ\} \cup \{F_{new}^\circ\}$	<p>refocusFilter</p>  <p>FC: $F_{old}^\circ \rightarrow F_{new}^\circ$</p>

Table 3.14: Operator `rejoin`

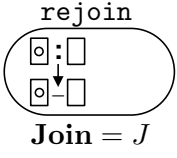
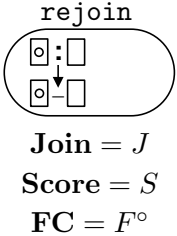
Operator Definition	Symbol
<u><code>rejoin(J)</code></u> Precondition: J is a set of join conditions between CoI_{src} and CoC_{src} , Postcondition: $JoinConds_{trg} = J, Scores_{trg} = Scores_{src},$ $ScoreFilters_{trg} = ScoreFilters_{src}$	
<u><code>rejoin(J, S, F°)</code></u> Precondition: J is a set of join conditions between CoI_{src} and CoC_{src} , S is a set of scores over CoI_{src} and CoC_{src} , and F are score filters over CoI_{src} and CoC_{src} with scores in S . Postcondition: $JoinConds_{trg} = J, Scores_{trg} = S, ScoreFilters_{trg} = F$	

Table 3.15: Operators `unrelate` and `untarget`

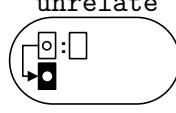
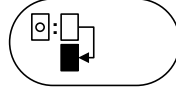
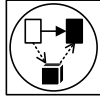
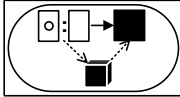
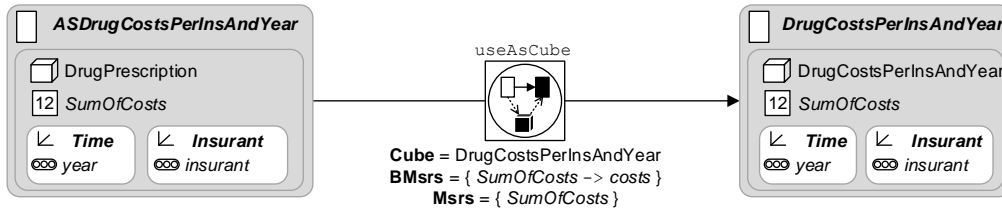
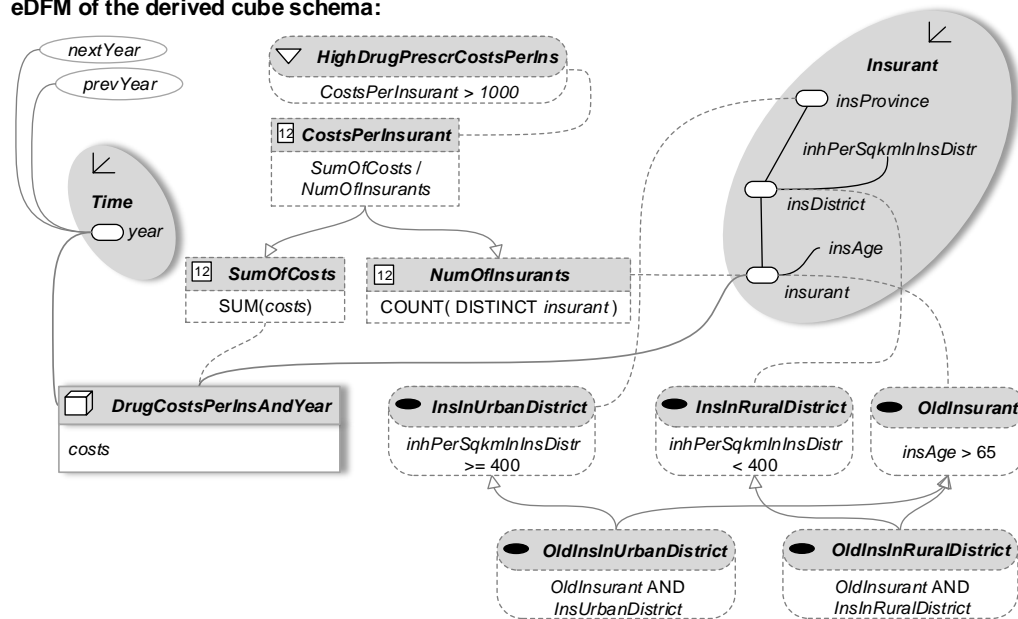
Operator Definition	Symbol
<u><code>unrelate()</code></u> Precondition: <i>No additional preconditions</i> Postcondition: $trg = CoC_{src}$	
<u><code>untarget()</code></u> Precondition: <i>No additional preconditions</i> Postcondition: $trg = CoI_{src}$	

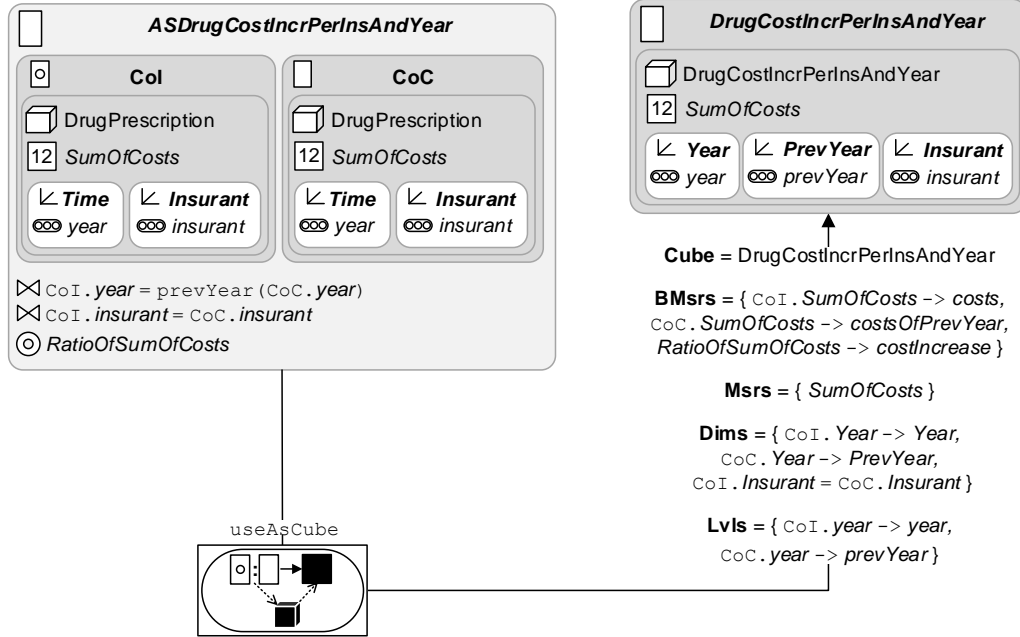
Table 3.16: Operator useAsCube

Operator Definition	Symbol
<p><u>useAsCube(C, \mathcal{B}, M)</u></p> <p>Precondition: src is a non-comparative analysis situation, C is a cube schema derived from src such that $BMsrsc = \mathcal{B}(Msrsrc)$ where \mathcal{B} is a base measure renaming from src to C, and M are measures applicable to $BMsrsc$.</p> <p>Postcondition: trg is a non-comparative analysis situation, $CubeSchema_{trg} = C$, $Msrstrg = M$, for all $D \in CubeSchema_C$, $DiceLvl_{trg}(D) = top_D$, $DiceNode_{trg}(D) = all_D$, $SliceConds_{trg}(D) = \emptyset$, $GranLvl_{trg}(D) = base_D$, and $FilterConds_{trg} = \emptyset$.</p>	<p>useAsCube</p>  <p>Cube = C BMsrsc = \mathcal{B} Msrsc = M</p>
<p><u>useAsCube($C, \mathcal{B}, \mathcal{D}, \mathcal{L}, \mathcal{A}, M$)</u></p> <p>Precondition: src is a comparative analysis situation, C is a cube schema derived from src such that $BMsrsc = \mathcal{B}(Msrsc_{CoIsrc} \dot{\cup} Msrsc_{CoIsrc} \dot{\cup} Scores_{src})$ where \mathcal{B} is a base measure renaming from src to C, and M are measures applicable to $BMsrsc$. Moreover, dimension schemas of $DimSchemas_C$ are renamed by \mathcal{D}, the levels are renamed by \mathcal{L}, and descriptive attributes by \mathcal{A}.</p> <p>Postcondition: trg is a non-comparative analysis situation, $CubeSchema_{trg} = C$, $Msrstrg = M$, for all $D \in CubeSchema_C$, $DiceLvl_{trg}(D) = top_D$, $DiceNode_{trg}(D) = all_D$, $SliceConds_{trg}(D) = \emptyset$, $GranLvl_{trg}(D) = base_D$, and $FilterConds_{trg} = \emptyset$.</p>	<p>useAsCube</p>  <p>Cube = C BMsrsc = \mathcal{M} Dims = \mathcal{D} Lvls = \mathcal{L} Attrs = \mathcal{A} Msrsc = M</p>

Navigation step:**SQL translation:**

```
create view DrugCostsPerInsAndYear as
select  year, insDistrict, SUM(costs) as costs
from    DrugPrescription natural join Time natural join Insurant
group by year, insDistrict
```

eDFM of the derived cube schema:Figure 3.3: Operator `useAsCube` (non-comparative analysis situation)

Navigation step:**SQL translation:**

```
create view DrugCostIncrPerInsAndYear as
```

```
select    Col.year as year, Col.insDistrict as insDistrict, Col.SumOfCosts as costs,
          CoC.year as prevYear, CoC.SumOfCosts as costsOfPrevYear,
          Col.SumOfCosts / CoC.SumOfCosts as costIncrease
```

```
from      (select    year, insDistrict, SUM(costs) as SumOfCosts
          from      DrugPrescription natural join Time natural join Insurant
          group by  year, insDistrict) Col
```

```
inner join (select    year, insDistrict, SUM(costs) as SumOfCosts
          from      DrugPrescription natural join Time natural join Insurant
          group by  year, insDistrict) CoC
```

```
on        Col.year = CoC.year + 1
```

Figure 3.4: Operator `useAsCube` (comparative analysis situation)

Chapter 4

Extension to Schema Level

4.1 Analysis Situation Schema

see Figure 4.1, 4.2

4.2 Navigation Step Schema

see Figure 4.3, 4.4, 4.5

4.3 Navigation guards

see Table 4.1, 4.2, 4.4, 4.5

see Figure 4.6, 4.7

Table 4.1: Navigation guard expressions concerning granularity levels (*as* denotes a non-comparative and *cas* a comparative analysis situation)

Operator definition
<u>$as.\text{granLevel}(D)$:</u> If $D \in \text{DimSchemas}_{as}$, then $as.\text{granLevel}(D) = \text{GranLvl}_{as}(D)$.
<u>$cas.\text{granLevelOfCoI}(D)$:</u> If $D \in \text{DimSchemas}_{CoI_{cas}}$, then $cas.\text{granLevelOfCoI}(D) = \text{GranLvl}_{CoI_{cas}}(D)$.
<u>$cas.\text{granLevelOfCoC}(D)$:</u> If $D \in \text{DimSchemas}_{CoC_{cas}}$, then $cas.\text{granLevelOfCoC}(D) = \text{GranLvl}_{CoC_{cas}}(D)$.
<u>$as.\text{hasSuperGranLevel}(D)$:</u> If $D \in \text{DimSchemas}_{as}$, then $as.\text{hasSuperGranLevel}(D)$ is true iff $(\exists G \in \text{Lvls}_D)(\text{GranLvl}_{as}(D) \rightarrow G)$
<u>$cas.\text{hasSuperGranLevelInCoI}(D)$:</u> If $D \in \text{DimSchemas}_{CoI_{cas}}$, then $cas.\text{hasSuperGranLevelInCoI}(D)$ is true iff $(\exists G \in \text{Lvls}_D)(\text{GranLvl}_{CoI_{cas}}(D) \rightarrow G)$
<u>$cas.\text{hasSuperGranLevelInCoC}(D)$:</u> If $D \in \text{DimSchemas}_{CoC_{cas}}$, then $cas.\text{hasSuperGranLevelInCoC}(D)$ is true iff $(\exists G \in \text{Lvls}_D)(\text{GranLvl}_{CoC_{cas}}(D) \rightarrow G)$
<u>$as.\text{hasSubGranLevel}(D)$:</u> If $D \in \text{DimSchemas}_{as}$, then $as.\text{hasSubGranLevel}(D)$ is true iff $(\exists G \in \text{Lvls}_D)(G \rightarrow \text{GranLvl}_{as}(D))$
<u>$cas.\text{hasSubGranLevelInCoI}(D)$:</u> If $D \in \text{DimSchemas}_{CoI_{cas}}$, then $cas.\text{hasSubGranLevelInCoI}(D)$ is true iff $(\exists G \in \text{Lvls}_D)(G \rightarrow \text{GranLvl}_{CoI_{cas}}(D))$
<u>$cas.\text{hasSubGranLevelInCoC}(D)$:</u> If $D \in \text{DimSchemas}_{CoC_{cas}}$, then $cas.\text{hasSubGranLevelInCoC}(D)$ is true iff $(\exists G \in \text{Lvls}_D)(G \rightarrow \text{GranLvl}_{CoC_{cas}}(D))$

Table 4.2: Navigation guard expressions concerning dice levels (*as* denotes a non-comparative and *cas* a comparative analysis situation)

Operator definition
<u>$as.diceLevel(D)$:</u> If $D \in DimSchemas_{as}$, then $as.diceLevel(D) = DiceLvl_{as}(D)$.
<u>$cas.diceLevelOfCoI(D)$:</u> If $D \in DimSchemas_{CoI_{cas}}$, then $cas.diceLevelOfCoI(D) = DiceLvl_{CoI_{cas}}(D)$.
<u>$cas.diceLevelOfCoC(D)$:</u> If $D \in DimSchemas_{CoC_{cas}}$, then $cas.diceLevelOfCoC(D) = DiceLvl_{CoC_{cas}}(D)$.
<u>$as.hasSuperDiceLevel(D)$:</u> If $D \in DimSchemas_{as}$, then $as.hasSuperDiceLevel(D)$ is true iff $(\exists L \in Lvl_D)(DiceLvl_{as}(D) \rightarrow L)$
<u>$cas.hasSuperDiceLevelInCoI(D)$:</u> If $D \in DimSchemas_{CoI_{cas}}$, then $cas.hasSuperDiceLevelInCoI(D)$ is true iff $(\exists L \in Lvl_D)(DiceLvl_{CoI_{cas}}(D) \rightarrow L)$
<u>$cas.hasSuperDiceLevelInCoC(D)$:</u> If $D \in DimSchemas_{CoC_{cas}}$, then $cas.hasSuperDiceLevelInCoC(D)$ is true iff $(\exists L \in Lvl_D)(DiceLvl_{CoC_{cas}}(D) \rightarrow L)$
<u>$as.hasSubDiceLevel(D)$:</u> If $D \in DimSchemas_{as}$, then $as.hasSubDiceLevel(D)$ is true iff $(\exists L \in Lvl_D)(L \rightarrow DiceLvl_{as}(D))$
<u>$cas.hasSubDiceLevelInCoI(D)$:</u> If $D \in DimSchemas_{CoI_{cas}}$, then $cas.hasSubDiceLevelInCoI(D)$ is true iff $(\exists L \in Lvl_D)(L \rightarrow DiceLvl_{CoI_{cas}}(D))$
<u>$cas.hasSubDiceLevelInCoC(D)$:</u> If $D \in DimSchemas_{CoC_{cas}}$, then $cas.hasSubDiceLevelInCoC(D)$ is true iff $(\exists L \in Lvl_D)(L \rightarrow DiceLvl_{CoC_{cas}}(D))$

Table 4.3: Navigation guard expressions concerning dice nodes (*as* denotes a non-comparative and *cas* a comparative analysis situation)

Operator definition
<u>$as.diceNode(D)$:</u> If $D \in DimSchemas_{as}$, then $as.diceNode(D) = DiceNode_{as}(D)$.
<u>$cas.diceNodeOfCoI(D)$:</u> If $D \in DimSchemas_{CoI_{cas}}$, then $cas.diceNodeOfCoI(D) = DiceNode_{CoI_{cas}}(D)$.
<u>$cas.diceNodeOfCoC(D)$:</u> If $D \in DimSchemas_{CoC_{cas}}$, then $cas.diceNodeOfCoC(D) = DiceNode_{CoC_{cas}}(D)$.
<u>$as.hasSuperDiceNode(D)$:</u> If $D \in DimSchemas_{as}$, then $as.hasSuperDiceNode(D)$ is true iff $(\exists N \in Nodes_{as}(D))(DiceNode_{as}(D) \rightarrow N)$
<u>$cas.hasSuperDiceNodeInCoI(D)$:</u> If $D \in DimSchemas_{CoI_{cas}}$, then $cas.hasSuperDiceNodeInCoI(D)$ is true iff $(\exists N \in Nodes_{CoI_{cas}}(D))(DiceNode_{CoI_{cas}}(D) \rightarrow N)$
<u>$cas.hasSuperDiceNodeInCoC(D)$:</u> If $D \in DimSchemas_{CoC_{cas}}$, then $cas.hasSuperDiceNodeInCoC(D)$ is true iff $(\exists N \in Nodes_{CoC_{cas}}(D))(DiceNode_{CoC_{cas}}(D) \rightarrow N)$
<u>$as.hasSubDiceNode(D)$:</u> If $D \in DimSchemas_{as}$, then $as.hasSubDiceNode(D)$ is true iff $(\exists N \in Nodes_{as}(D))(N \rightarrow DiceNode_{as}(D))$
<u>$cas.hasSubDiceNodeInCoI(D)$:</u> If $D \in DimSchemas_{CoI_{cas}}$, then $cas.hasSubDiceNodeInCoI(D)$ is true iff $(\exists N \in Nodes_{CoI_{cas}}(D))(N \rightarrow DiceNode_{CoI_{cas}}(D))$
<u>$cas.hasSubDiceNodeInCoC(D)$:</u> If $D \in DimSchemas_{CoC_{cas}}$, then $cas.hasSubDiceNodeInCoC(D)$ is true iff $(\exists N \in Nodes_{CoC_{cas}}(D))(N \rightarrow DiceNode_{CoC_{cas}}(D))$

Table 4.4: Navigation guard expressions examining next and previous dice node (*as* denotes a non-comparative and *cas* a comparative analysis situation)

Operator definition
<p><u><i>as.hasNextDiceNode(D)</i></u>:</p> <p>If $D \in DimSchemas_{as}$, then <i>as.hasNextDiceNode(D)</i> is true iff $(\exists \hat{N}, N' \in Nodes_{as}(D))(\hat{N} \neq N' \wedge DiceNode_{as}(D) \rightarrow \hat{N} \wedge N' \rightarrow \hat{N} \wedge next(DiceNode_{as}(D)) = N')$</p>
<p><u><i>cas.hasNextDiceNodeInCoI(D)</i></u>:</p> <p>If $D \in DimSchemas_{CoI_{cas}}$, then <i>cas.hasNextDiceNodeInCoI(D)</i> is true iff $(\exists \hat{N}, N' \in Nodes_{CoI_{cas}}(D))(\hat{N} \neq N' \wedge DiceNode_{CoI_{cas}}(D) \rightarrow \hat{N} \wedge N' \rightarrow \hat{N} \wedge next(DiceNode_{CoI_{cas}}(D)) = N')$</p>
<p><u><i>cas.hasNextDiceNodeInCoC(D)</i></u>:</p> <p>If $D \in DimSchemas_{CoC_{cas}}$, then <i>cas.hasNextDiceNodeInCoC(D)</i> is true iff $(\exists \hat{N}, N' \in Nodes_{CoC_{cas}}(D))(\hat{N} \neq N' \wedge DiceNode_{CoC_{cas}}(D) \rightarrow \hat{N} \wedge N' \rightarrow \hat{N} \wedge next(DiceNode_{CoC_{cas}}(D)) = N')$</p>
<p><u><i>as.hasPrevDiceNode(D)</i></u>:</p> <p>If $D \in DimSchemas_{as}$, then <i>as.hasPrevDiceNode(D)</i> is true iff $(\exists \hat{N}, N' \in Nodes_{as}(D))(\hat{N} \neq N' \wedge DiceNode_{as}(D) \rightarrow \hat{N} \wedge N' \rightarrow \hat{N} \wedge prev(DiceNode_{as}(D)) = N')$</p>
<p><u><i>cas.hasPrevDiceNodeInCoI(D)</i></u>:</p> <p>If $D \in DimSchemas_{CoI_{cas}}$, then <i>cas.hasPrevDiceNodeInCoI(D)</i> is true iff $(\exists \hat{N}, N' \in Nodes_{CoI_{cas}}(D))(\hat{N} \neq N' \wedge DiceNode_{CoI_{cas}}(D) \rightarrow \hat{N} \wedge N' \rightarrow \hat{N} \wedge prev(DiceNode_{CoI_{cas}}(D)) = N')$</p>
<p><u><i>cas.hasPrevDiceNodeInCoC(D)</i></u>:</p> <p>If $D \in DimSchemas_{CoC_{cas}}$, then <i>cas.hasPrevDiceNodeInCoC(D)</i> is true iff $(\exists \hat{N}, N' \in Nodes_{CoC_{cas}}(D))(\hat{N} \neq N' \wedge DiceNode_{CoC_{cas}}(D) \rightarrow \hat{N} \wedge N' \rightarrow \hat{N} \wedge prev(DiceNode_{CoC_{cas}}(D)) = N')$</p>

Table 4.5: Navigation guard expressions examining slice conditions, filter conditions, score filter conditions, and whether a result set is empty (*as* denotes a non-comparative and *cas* a comparative analysis situation)

Operator definition
<u><i>as.containsSliceCond(D, P):</i></u> If $D \in DimSchemas_{as}$ and $P \in DimPredicates_D$, then <i>as.containsSliceCond(D, P)</i> is true iff $(\exists P' \in SliceConds_{as})(P \Rightarrow P')$
<u><i>cas.containsSliceCondInCoI(D, P):</i></u> If $D \in DimSchemas_{CoI_{cas}}$ and $P \in DimPredicates_D$, then <i>cas.containsSliceCondInCoI(D, P)</i> is true iff $(\exists P' \in SliceConds_{CoI_{cas}})(P \Rightarrow P')$
<u><i>cas.containsSliceCondInCoC(D, P):</i></u> If $D \in DimSchemas_{CoC_{cas}}$ and $P \in DimPredicates_D$, then <i>cas.containsSliceCondInCoC(D, P)</i> is true iff $(\exists P' \in SliceConds_{CoC_{cas}})(P \Rightarrow P')$
<u><i>as.containsBMsrCond(F):</i></u> If $B \in BMsrPredicates_{as}$, then <i>as.containsBMsrCond(B)</i> is true iff $(\exists B' \in BMsrConds_{as})(B \Rightarrow B')$
<u><i>cas.containsBMsrCondInCoI(B):</i></u> If $B \in BMsrPredicates_{CoI_{cas}}$, then <i>cas.containsBMsrCondInCoI(F)</i> is true iff $(\exists B' \in BMsrConds_{CoI_{cas}})(B \Rightarrow B')$
<u><i>cas.containsBMsrCondInCoC(B):</i></u> If $B \in BMsrPredicates_{CoC_{cas}}$, then <i>cas.containsBMsrCondInCoC(F)</i> is true iff $(\exists B' \in BMsrConds_{CoC_{cas}})(B \Rightarrow B')$
<u><i>as.containsFilterCond(F):</i></u> If $F \in MsrPredicates_{as}$, then <i>as.containsFilterCond(F)</i> is true iff $(\exists F' \in FilterConds_{as})(F \Rightarrow F')$
<u><i>cas.containsFilterCondInCoI(F):</i></u> If $F \in MsrPredicates_{CoI_{cas}}$, then <i>cas.containsFilterCondInCoI(F)</i> is true iff $(\exists F' \in FilterConds_{CoI_{cas}})(F \Rightarrow F')$
<u><i>cas.containsFilterCondInCoC(F):</i></u> If $F \in MsrPredicates_{CoC_{cas}}$, then <i>cas.containsFilterCondInCoC(F)</i> is true iff $(\exists F' \in FilterConds_{CoC_{cas}})(F \Rightarrow F')$
<u><i>cas.containsScoreFilterCond(F):</i></u> If F is a score filter over CoI_{cas} and CoC_{cas} , then <i>cas.containsScoreFilterCond(F)</i> is true iff $(\exists F' \in ScoreFilters_{cas})(F \Rightarrow F')$
<u><i>as.hasEmptyResult()</i> or <i>cas.hasEmptyResult():</i></u> <i>as.hasEmptyResult()</i> (<i>cas.hasEmptyResult()</i>) is true iff <i>as</i> (<i>cas</i>) has an empty result set.

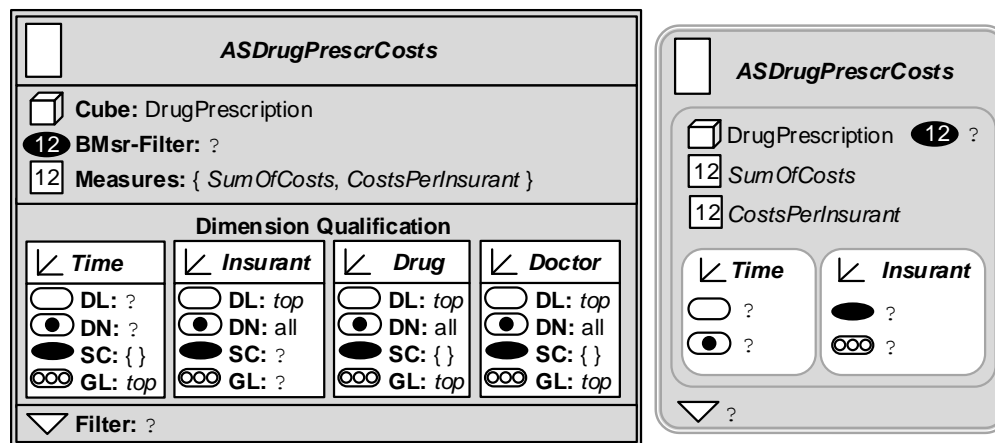


Figure 4.1: Non-comparative analysis situation with unbound variables

CompDrugPrescrCosts							
Context of Interest (CoI)				Context of Comparison (CoC)			
<input type="checkbox"/> Cube: DrugPrescription <input checked="" type="radio"/> BMSr-Filter: ? <input type="checkbox"/> Measures: { SumOfCosts, CostsPerInsurant }				<input type="checkbox"/> Cube: DrugPrescription <input checked="" type="radio"/> BMSr-Filter: ? <input type="checkbox"/> Measures: { SumOfCosts, CostsPerInsurant }			
Dimension Qualification				Dimension Qualification			
Time	Insurant	Drug	Doctor	Time	Insurant	Drug	Doctor
<input type="radio"/> DL: ?	<input type="radio"/> DL: top	<input type="radio"/> DL: top	<input type="radio"/> DL: top	<input type="radio"/> DL: ?	<input type="radio"/> DL: top	<input type="radio"/> DL: top	<input type="radio"/> DL: top
<input checked="" type="radio"/> DN: ?	<input checked="" type="radio"/> DN: all	<input checked="" type="radio"/> DN: all	<input checked="" type="radio"/> DN: all	<input checked="" type="radio"/> DN: ?	<input checked="" type="radio"/> DN: all	<input checked="" type="radio"/> DN: all	<input checked="" type="radio"/> DN: all
<input checked="" type="radio"/> SC: { }	<input checked="" type="radio"/> SC: ?	<input checked="" type="radio"/> SC: { }	<input checked="" type="radio"/> SC: { }	<input checked="" type="radio"/> SC: { }	<input checked="" type="radio"/> SC: ?	<input checked="" type="radio"/> SC: { }	<input checked="" type="radio"/> SC: { }
<input checked="" type="radio"/> GL: top	<input checked="" type="radio"/> GL: ?	<input checked="" type="radio"/> GL: top	<input checked="" type="radio"/> GL: top	<input checked="" type="radio"/> GL: top	<input checked="" type="radio"/> GL: ?	<input checked="" type="radio"/> GL: top	<input checked="" type="radio"/> GL: top
<input type="checkbox"/> Filter: ?				<input type="checkbox"/> Filter: { }			
<input checked="" type="checkbox"/> Join Condition: { CoI.insProvince = CoC.insProvince, CoI.topTime = CoC.topTime, CoI.topDrug = CoC.topDrug, CoI.topDoctor = CoC.topDoctor }							
<input checked="" type="radio"/> Scores: { RatioOfSumOfCosts, RatioOfCostsPerInsurant }							
<input checked="" type="radio"/> Score Filter: ?							

CompDrugPrescrCosts	
<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <input type="checkbox"/> Context Of Interest (CoI) <input type="checkbox"/> DrugPrescription <input checked="" type="radio"/> ? <input type="checkbox"/> SumOfCosts <input type="checkbox"/> CostsPerInsurant <div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid #ccc; padding: 2px;"> Time <input type="radio"/> ? <input checked="" type="radio"/> ? </div> <div style="border: 1px solid #ccc; padding: 2px;"> Insurant <input checked="" type="radio"/> ? <input checked="" type="radio"/> ? </div> </div> <input type="checkbox"/> ? </div>	<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <input type="checkbox"/> Context Of Comparison (CoC) <input type="checkbox"/> DrugPrescription <input checked="" type="radio"/> ? <input type="checkbox"/> SumOfCosts <input type="checkbox"/> CostsPerInsurant <div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid #ccc; padding: 2px;"> Time <input type="radio"/> ? <input checked="" type="radio"/> ? </div> <div style="border: 1px solid #ccc; padding: 2px;"> Insurant <input checked="" type="radio"/> ? <input checked="" type="radio"/> ? </div> </div> <input type="checkbox"/> ? </div>
<input checked="" type="checkbox"/> CoI.insProvince = CoC.insProvince <input checked="" type="radio"/> RatioOfSumOfCosts <input checked="" type="radio"/> RatioOfCostsPerInsurant <input checked="" type="radio"/> ?	

Figure 4.2: Comparative analysis situation with unbound variables

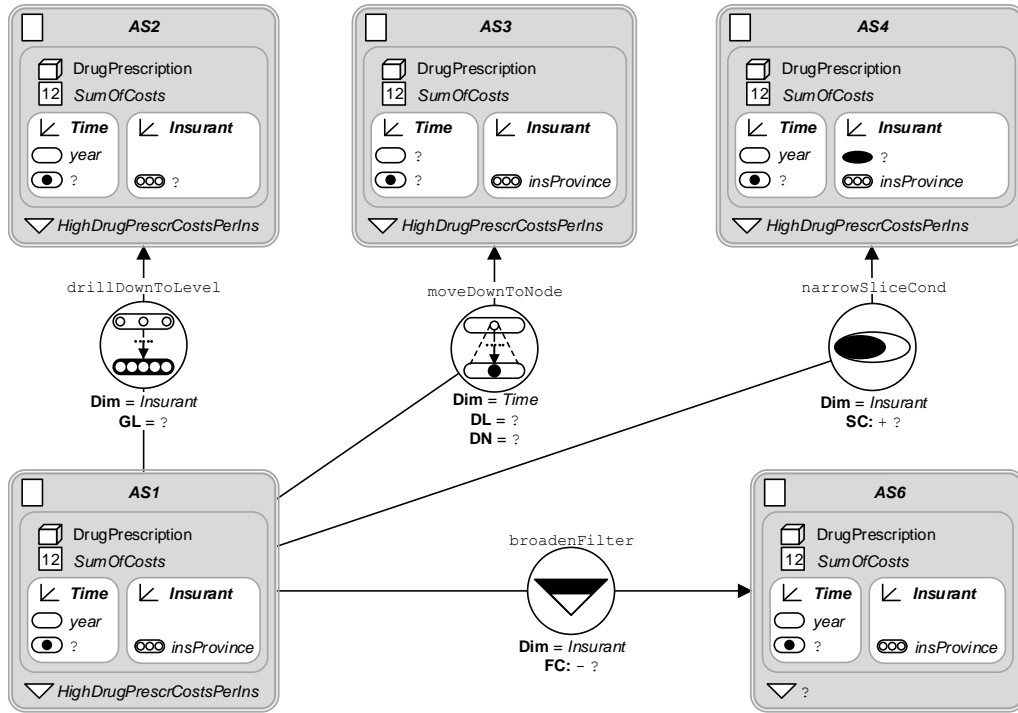


Figure 4.3: Navigation steps with unbound variables

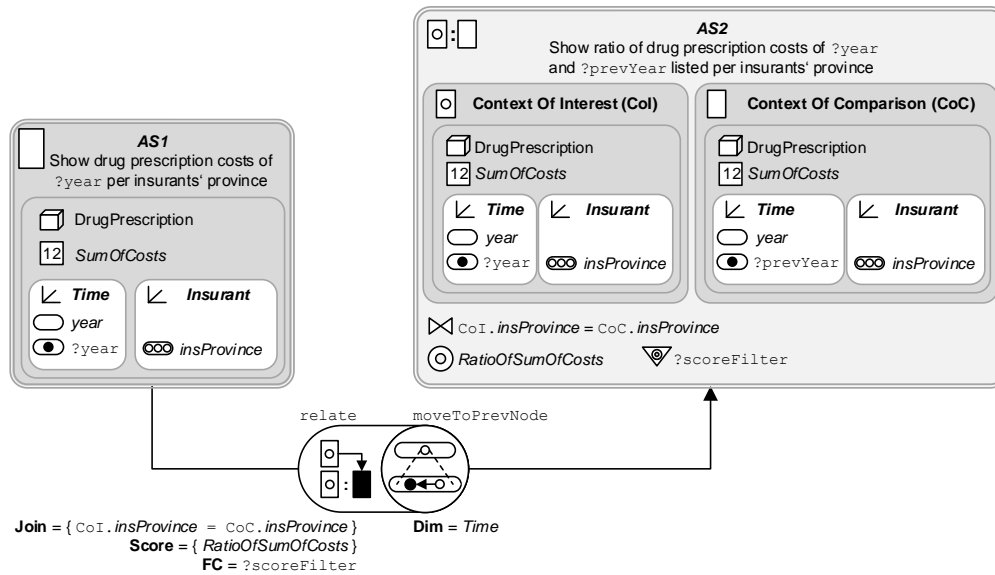


Figure 4.4: Navigation steps with unbound variables involving a comparative analysis situation

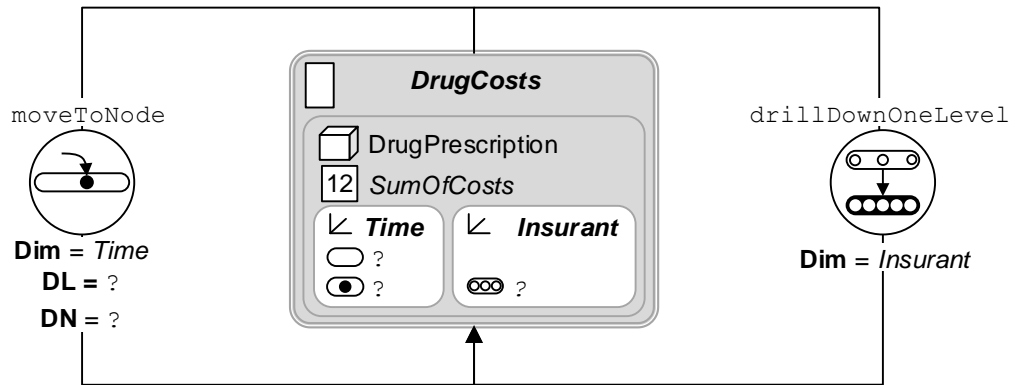


Figure 4.5: Navigation step schemas with equal source and targets

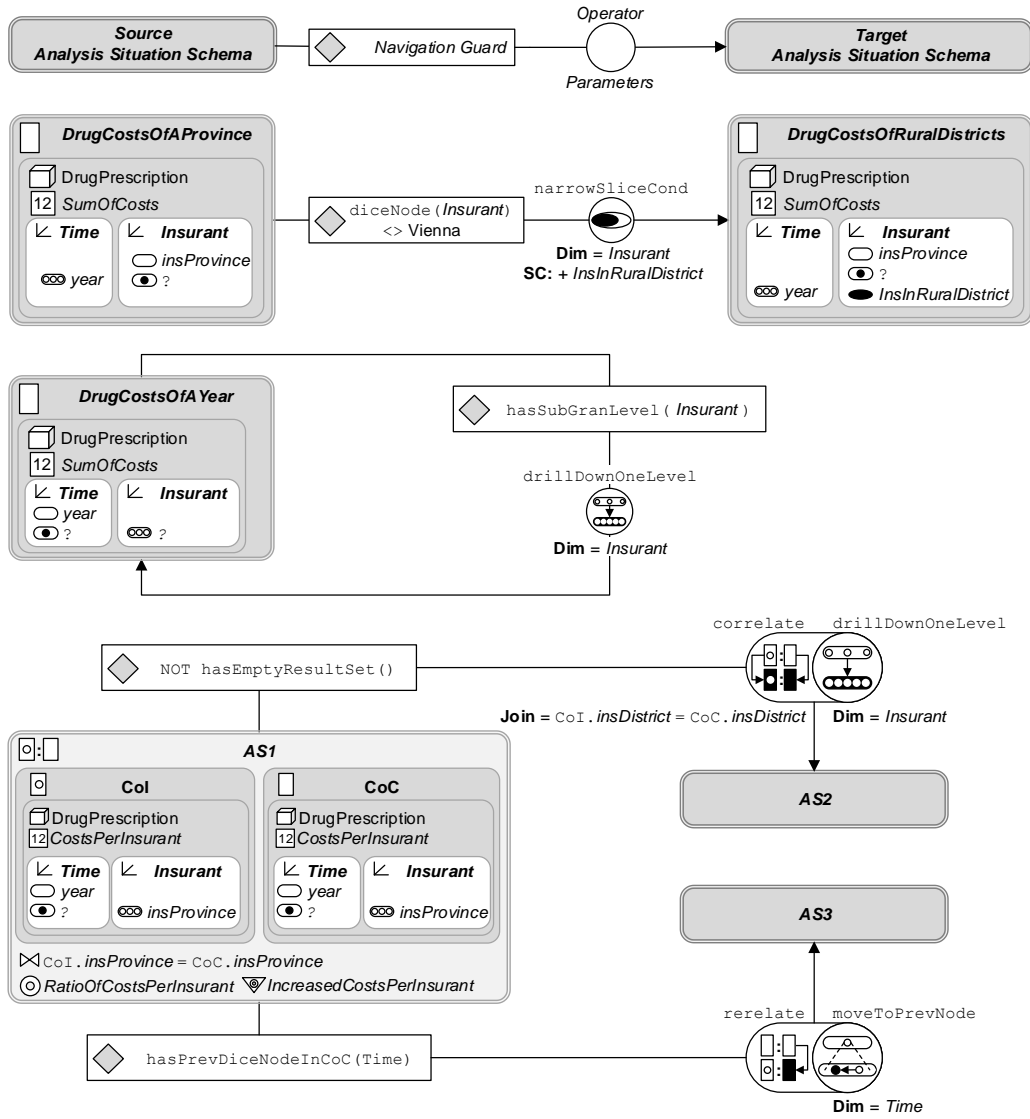
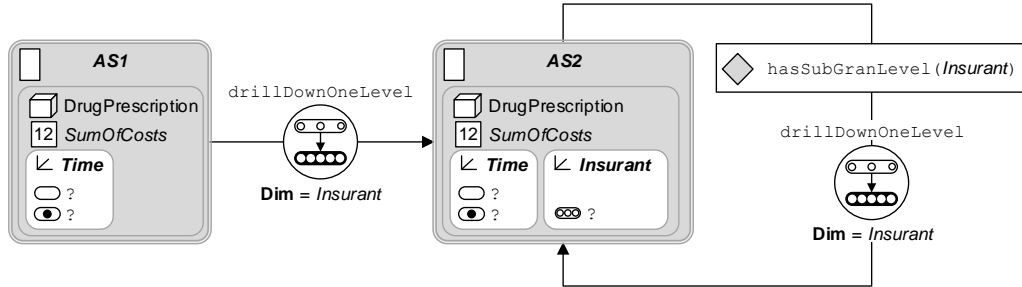


Figure 4.6: Navigation guards

Navigation pattern for refining granularity level of *Insurant*



Navigation pattern for iterating over all years

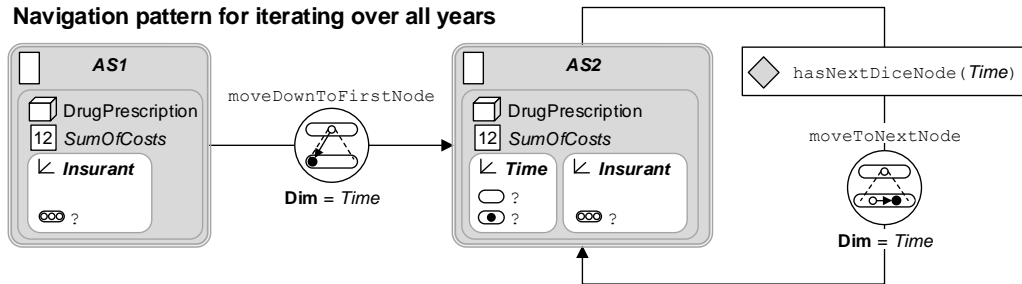


Figure 4.7: Navigation patterns

Chapter 5

Business Intelligence (BI) Analysis Graphs

5.1 Definition of BI Analysis Graphs

see Figure 5.1

5.2 BI Analysis Graph Schemas

see Figure 5.2

5.3 Instances of Analysis Graph Schemas

see Figure 5.3

5.4 Composite analysis situation

see Figure 5.4, 5.5, 5.6

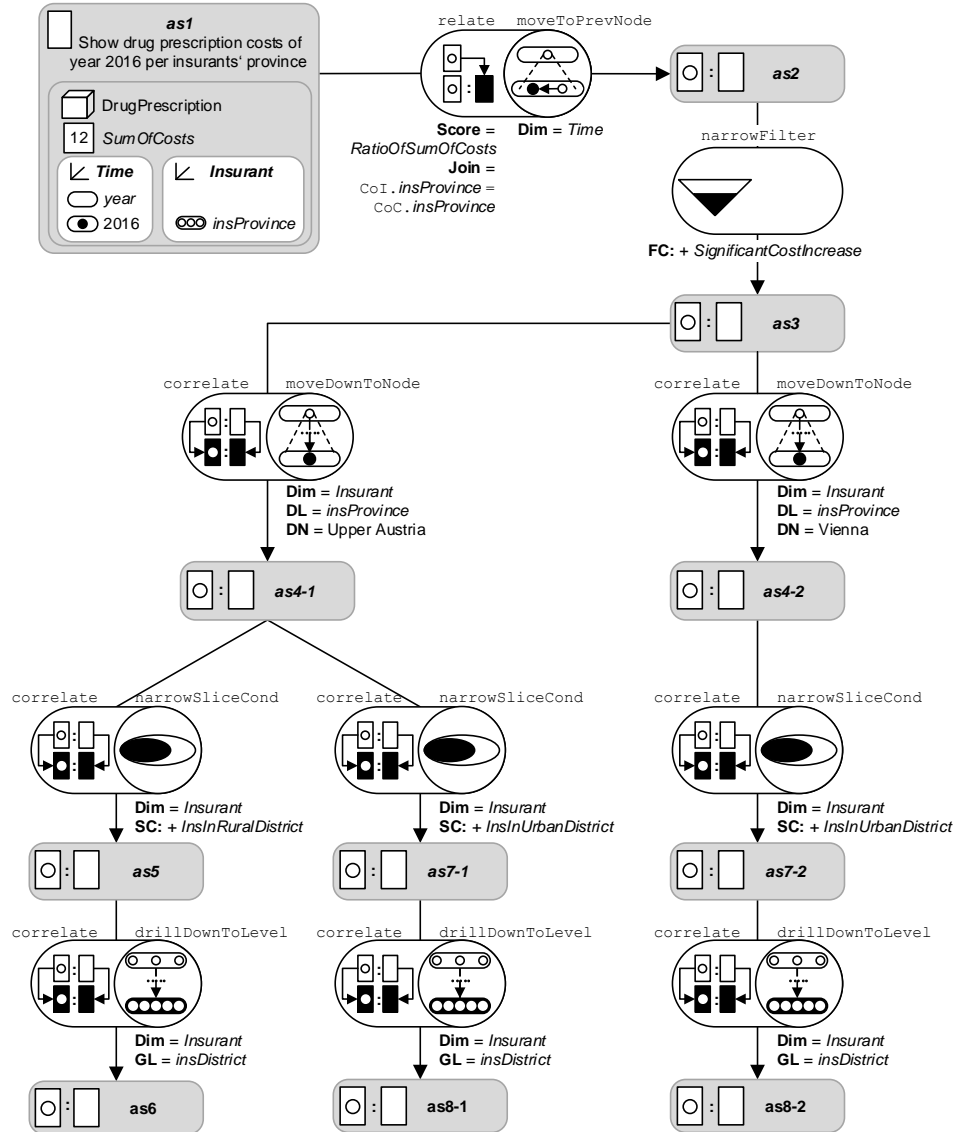


Figure 5.1: Example of BI analysis graph extracted from the running example

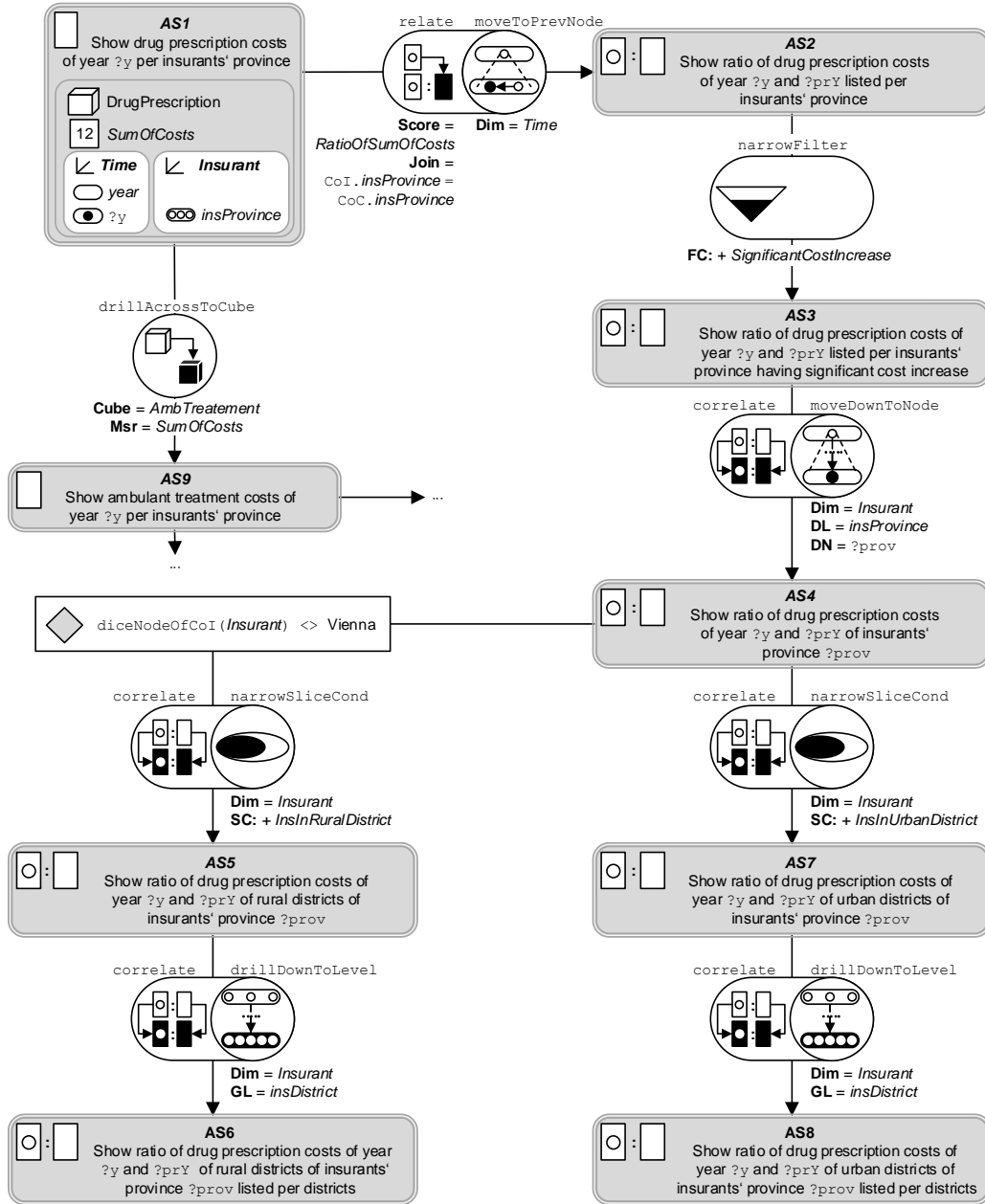


Figure 5.2: Example of BI analysis graph schema extracted from the running example

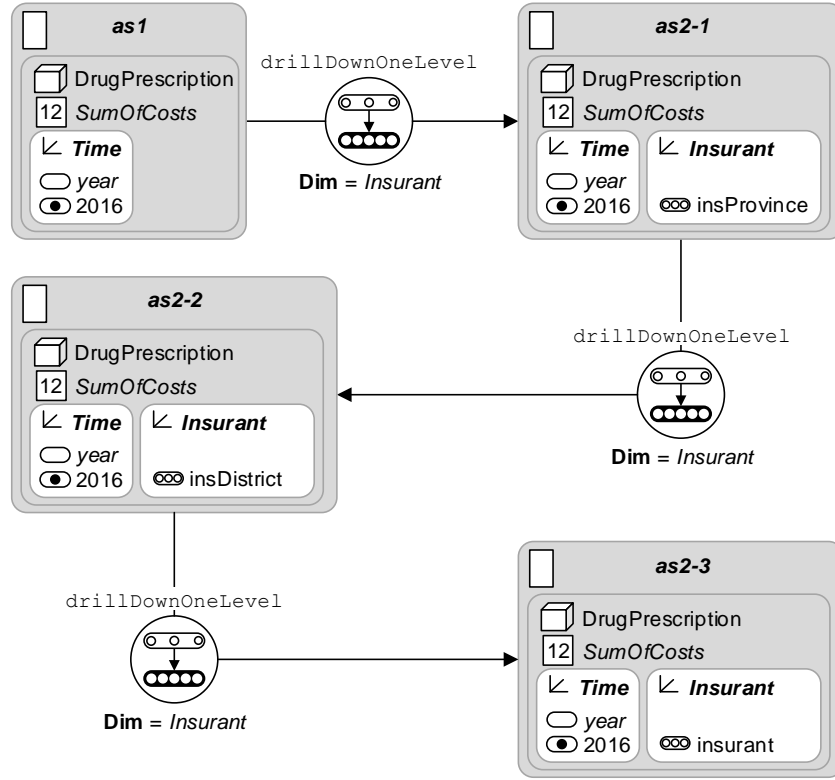


Figure 5.3: Analysis graph instance of analysis graph schema of the first navigation pattern of Figure 4.7

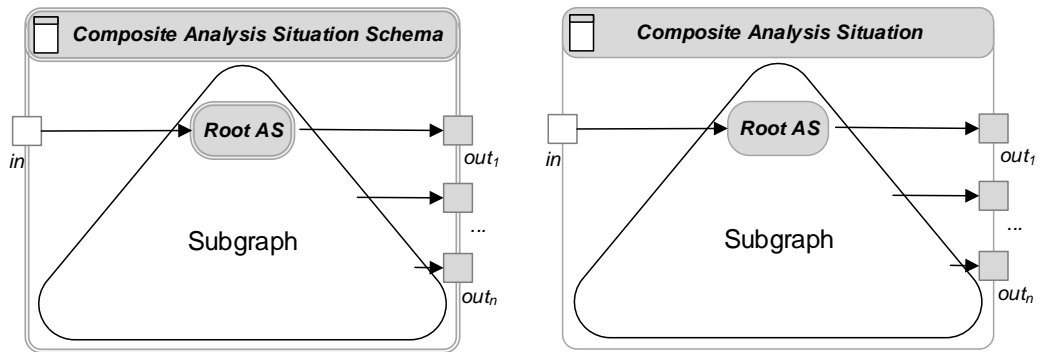


Figure 5.4: Composite analysis situation (schema)

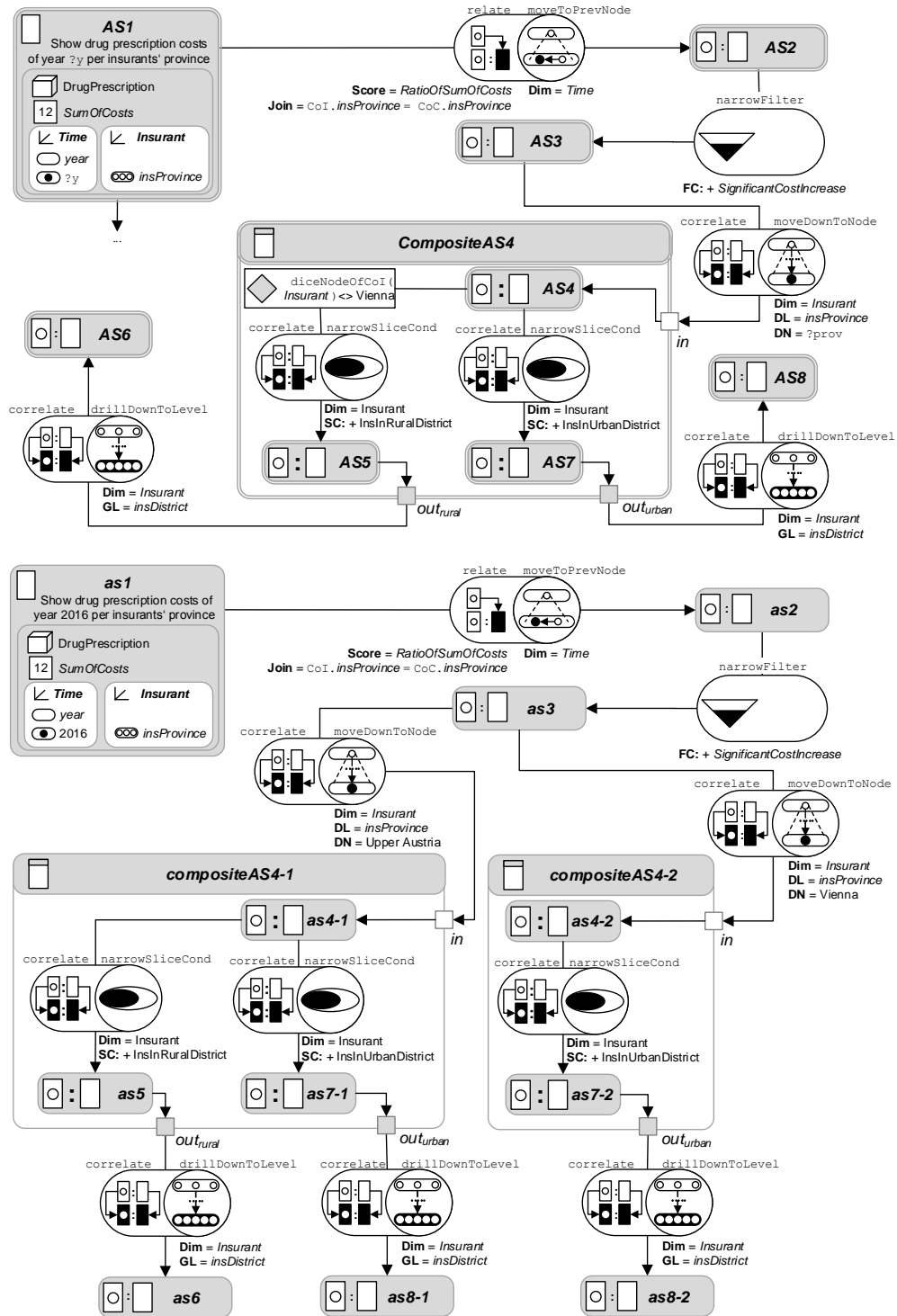


Figure 5.5: Composite analysis situation (schema) — running example

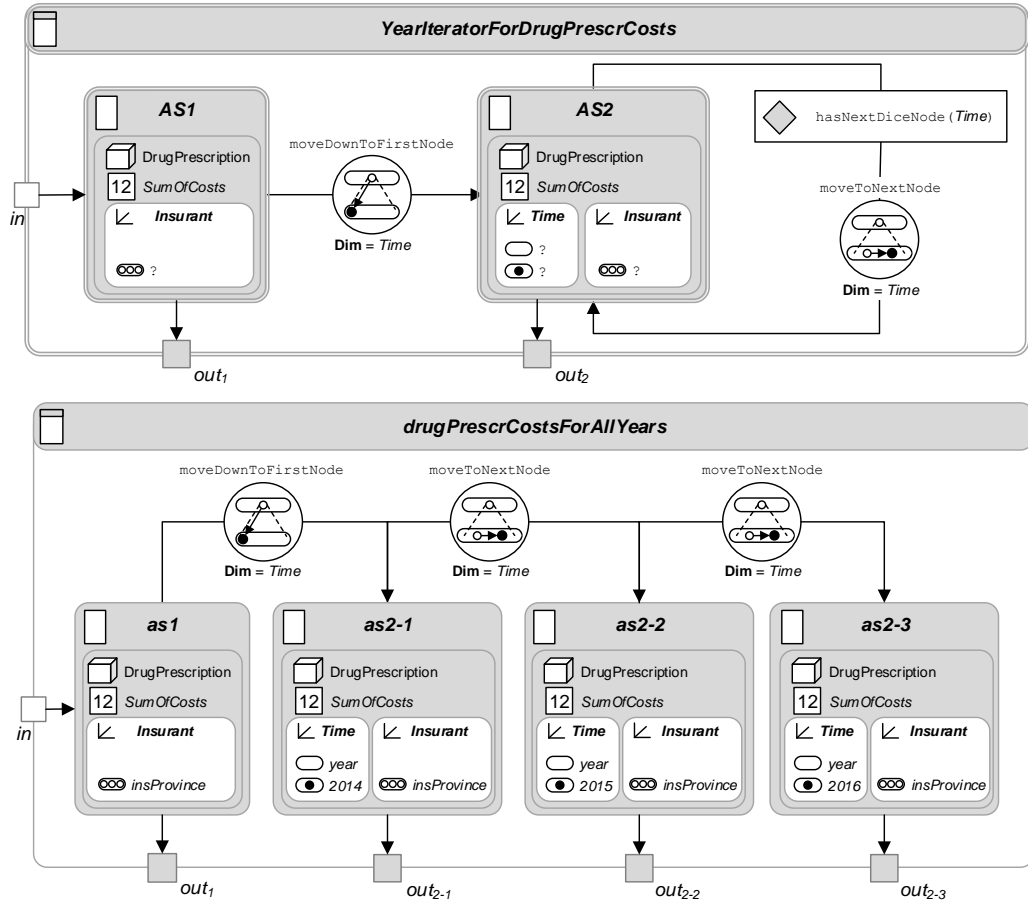


Figure 5.6: Composite analysis situation (schema) example (navigation pattern)

List of Figures

1.1	Health care use case – overview of an analysis process	2
1.2	eDFM for Drug Prescriptions	3
1.3	eDFM for Ambulant Treatment and Hospitalization	4
1.4	Cube Instance	5
2.1	Graphical Template of a Non-comparative Analysis Situation .	8
2.2	Example of a Graphical Representation of a Non-comparative Analysis Situation	8
2.3	Example of an Alternative Graphical Representation of a Non- comparative Analysis Situation	9
2.4	SQL Translation	10
2.5	Example of an SQL Translation	10
2.6	Graphical Template of a Comparative Analysis Situation . . .	11
2.7	Example of a Graphical Representation of a Comparative Ana- lysis Situation	12
2.8	Example of an Alternative Graphical Representation of a Com- parative Analysis Situation	13
2.9	SQL Translation of a Comparative Analysis Situation	14
2.10	Example of an SQL Translation of a Comparative Analysis Situation	15
3.1	Navigation operator examples	18
3.2	Example of a comparative navigation step	24
3.3	Operator <code>useAsCube</code> (non-comparative analysis situation) . . .	35
3.4	Operator <code>useAsCube</code> (comparative analysis situation)	36

4.1	Non-comparative analysis situation with unbound variables . .	43
4.2	Comparative analysis situation with unbound variables	44
4.3	Navigation steps with unbound variables	45
4.4	Navigation steps with unbound variables involving a compar- ative analysis situation	45
4.5	Navigation step schemas with equal source and targets	46
4.6	Navigation guards	47
4.7	Navigation patterns	48
5.1	Example of BI analysis graph extracted from the running ex- ample	50
5.2	Example of BI analysis graph schema extracted from the run- ning example	51
5.3	Analysis graph instance of analysis graph schema of the first navigation pattern of Figure 4.7	52
5.4	Composite analysis situation (schema)	52
5.5	Composite analysis situation (schema) — running example . .	53
5.6	Composite analysis situation (schema) example (navigation pattern)	54

List of Tables

3.1	Operators <code>drillDownOneLevel</code> , <code>drillDownToLevel</code> , <code>rollUpOneLevel</code> , and <code>rollUpToLevel</code>	19
3.2	Operators <code>moveDownToNode</code> , <code>moveUpToNode</code> , <code>moveAsideToNode</code> , and <code>moveToNode</code>	20
3.3	Operators <code>moveDownToFirstNode</code> , <code>moveDownToLastNode</code> , <code>moveToNextNode</code> , and <code>moveToPrevNode</code>	21
3.4	Operators <code>narrowSliceCond</code> , <code>broadenSliceCond</code> , and <code>refocusSliceCond</code>	22
3.5	Operators <code>narrowBMsCond</code> , <code>broadenBMsCond</code> , and <code>refocusBMsCond</code>	23
3.6	Operators <code>addMeasure</code> , <code>dropMeasure</code> , <code>refocusMeasure</code> , <code>moveDownToMeasure</code> , and <code>moveUpToMeasure</code>	25
3.7	Operator <code>narrowFilter</code>	26
3.8	Operator <code>drillAcrossToCube</code>	27
3.9	Operators <code>relate</code> and <code>target</code>	28
3.10	Operator <code>rerelate</code>	29
3.11	Operator <code>retarget</code>	30
3.12	Operator <code>correlate</code>	31
3.13	Operators <code>narrowFilter</code> , <code>broadenFilter</code> , and <code>refocusFilter</code> for score filters	32
3.14	Operator <code>rejoin</code>	33
3.15	Operators <code>unrelate</code> and <code>untarget</code>	33
3.16	Operator <code>useAsCube</code>	34

4.1	Navigation guard expressions concerning granularity levels (<i>as</i> denotes a non-comparative and <i>cas</i> a comparative analysis situation)	38
4.2	Navigation guard expressions concerning dice levels (<i>as</i> denotes a non-comparative and <i>cas</i> a comparative analysis situation)	39
4.3	Navigation guard expressions concerning dice nodes (<i>as</i> denotes a non-comparative and <i>cas</i> a comparative analysis situation)	40
4.4	Navigation guard expressions examining next and previous dice node (<i>as</i> denotes a non-comparative and <i>cas</i> a comparative analysis situation)	41
4.5	Navigation guard expressions examining slice conditions, filter conditions, score filter conditions, and whether a result set is empty (<i>as</i> denotes a non-comparative and <i>cas</i> a comparative analysis situation)	42