CineCubes: Cubes as Movie Stars with Little Effort

Dimitrios Gkesoulis*

UTC Creative Lab

Ioannina, Hellas

Panos Vassiliadis

Dept. of Computer Science & Engineering,

Univ. Ioannina, Hellas

*work conducted while in the Univ. Ioannina

Univ. of Ioannina

Can we answer user queries with more than just a set of records?

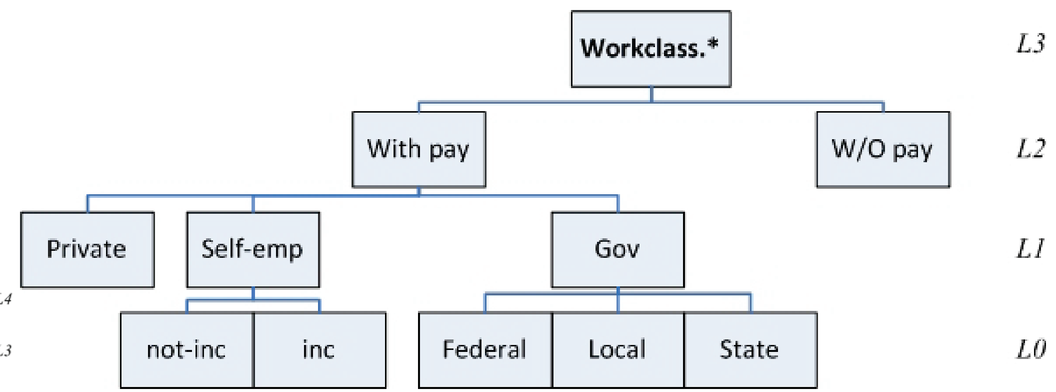
We should and can produce query results that are

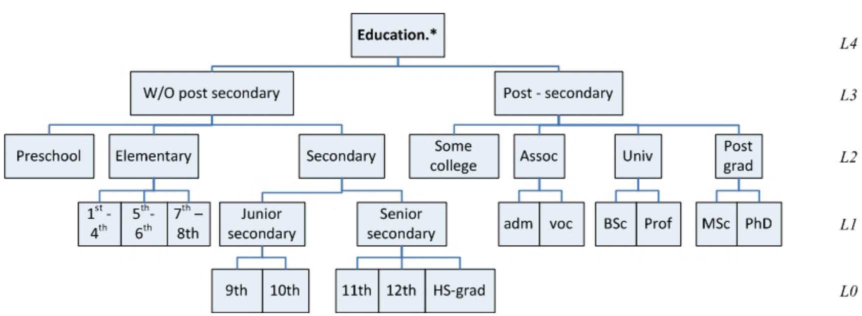
- *properly visualized,*
- *enriched with textual comments,*
- *vocally enriched,*

… but then, you have a movie

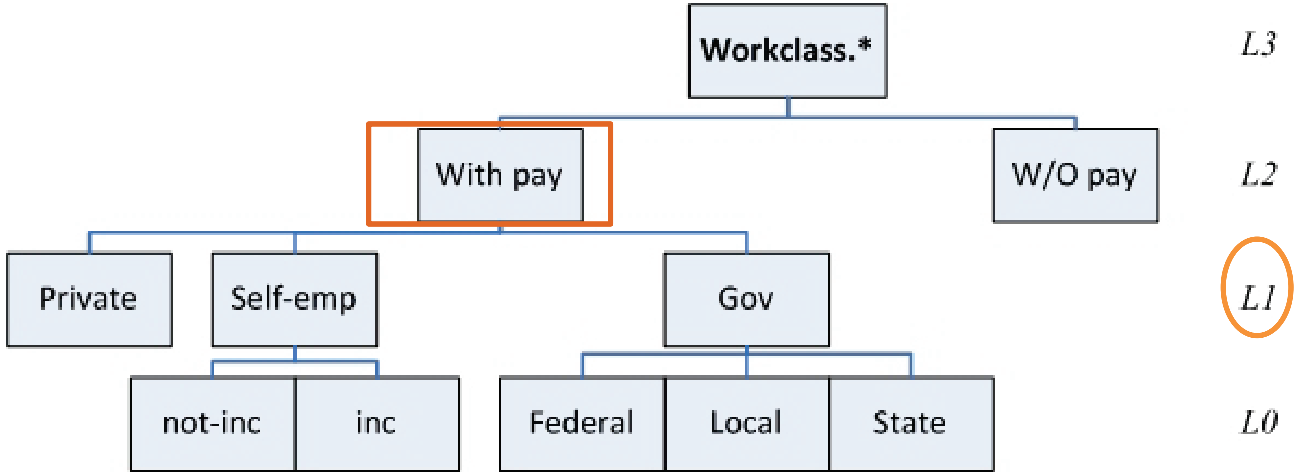
Example

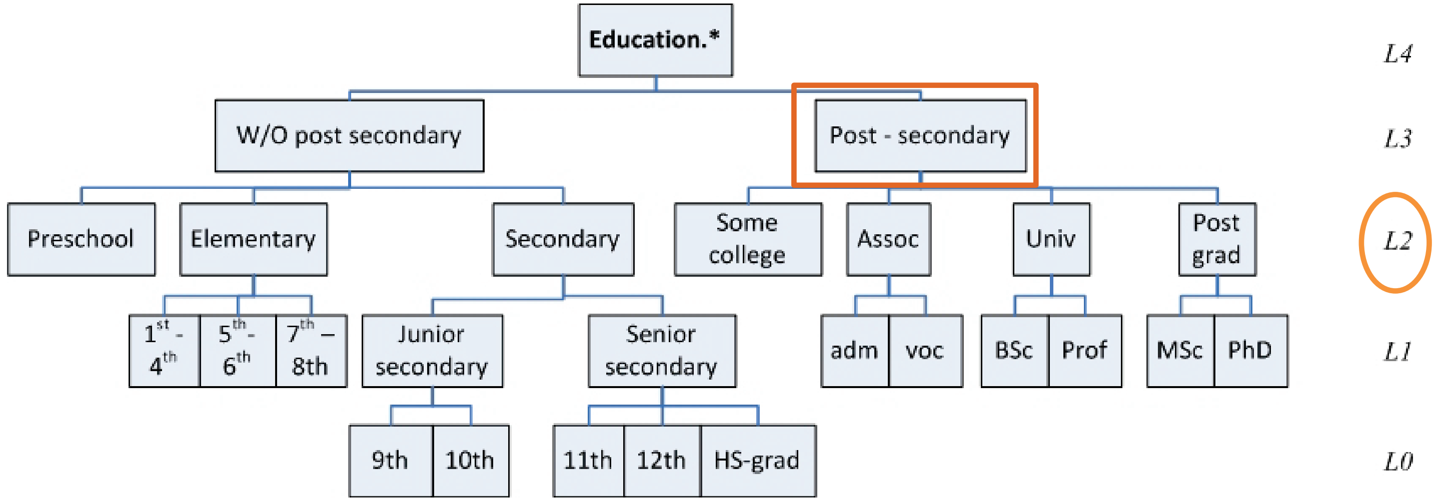
- Find the average work hours per week *//measure*
  - For persons with *//selection conditions*
    - work_class.level2='With-Pay' , and
    - education.level3= 'Post-Sec'
  - Grouped per *//groupers*
    - work_class.level1
    - education.level3



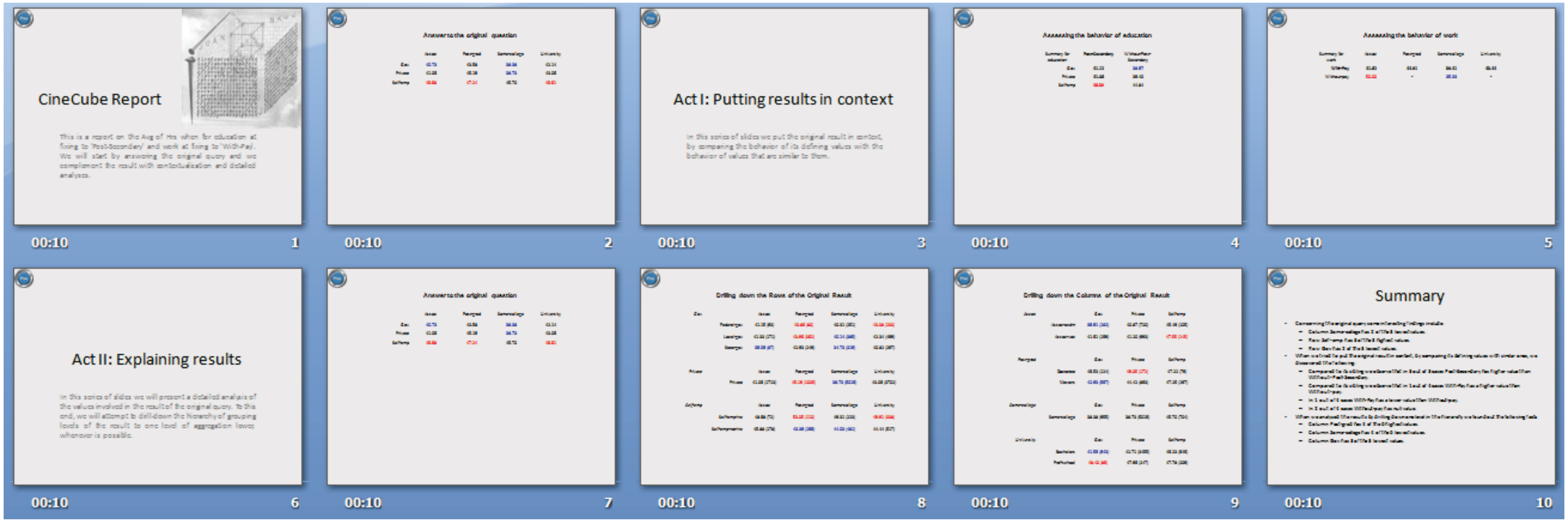


Example


Workclass.*
With pay
W/O pay
Private
Self-emp
Gov
not-inc
inc
Federal
Local
State
L3
L2
L1
L0


Education.*
W/O post secondary
Post - secondary
Preschool
Elementary
Secondary
Some college
Assoc
Univ
Post grad
1st - 4th
5th - 6th
7th – 8th
Junior secondary
Senior secondary
adm
voc
BSc
Prof
MSc
PhD
9th
10th
11th
12th
HS-grad
L4
L3
L2
L1
L0

Example: Result



Answer to the original question

	Assoc	Post-grad	Some-college	University
Gov	40.73	43.58	38.38	42.14
Private	41.06	45.19	38.73	43.06
Self-emp	46.68	47.24	45.70	46.61

Here, you can see the answer of the original query. You have specified education to be equal to 'Post-Secondary', and work to be equal to 'With-Pay'. We report on Avg of work hours per week grouped by education at level 2, and work at level 1 .

You can observe the results in this table. We highlight the largest values with red and the lowest values with blue color.

Column Some-college has 2 of the 3 lowest values.

Row Self-emp has 3 of the 3 highest values.

Row Gov has 2 of the 3 lowest values.

1

	Assoc	Post-grad	Some-college	University
Gov	40.73	43.58	38.38	42.14
Private	41.06	45.19	38.73	43.06
Self-emp	46.68	47.24	45.70	46.61

Original query

Here, you can see the answer of the original query. You have specified education to be equal to 'Post-Secondary', and work to be equal to 'With-Pay'. We report on Avg of Hrs grouped by education at level 2, and work at level 1. We highlight the largest values with red and the lowest values with blue.
Column Some-college has 2 of the 3 lowest values.
Row Self-emp has 3 of the 3 highest values.
Row Gov has 2 of the 3 lowest values.

2

	Post-Secondary	Without Post-Secondary
Gov	**41.12**	38.97
Private	**41.06**	39.40
Self-emp	**46.39**	44.84

Summary for education

In this graphic, we put the original request in context by comparing the value 'Post-Secondary' for education at level 3 with its sibling values. We calculate the Avg of Hrs while fixing education at level 4 to be equal to "ALL", and work at level 2 to be equal to "With-Pay'. We highlight the reference cells with bold, the highest value with red and the lowest value with blue.
Compared to its sibling we observe that in 3 out of 3 cases Post-Secondary has higher value than Without-Post-Secondary.

Act I (sl. 2,3)

In this slide, we drill-down one level for all values of dimension work at level 0. For each cell we show both the Avg of Hrs and the number of tuples that correspond to it in parentheses. ...
Column Post-grad has 4 of the 6 highest values.
Column Some-college has 4 of the 6 lowest values.

3

	Assoc	Post-grad	Some-college	University
With-Pay	**41.62**	**44.91**	**39.41**	**43.44**
Without-pay	50.00	-	35.33	-

Summary for work

Drilling down education

Act II (sl. 3,4)

5

Assoc		Gov	Private	Self-emp
	Assoc-acdm	39.91 (182)	40.87 (720)	45.49 (105)
	Assoc-voc	41.61 (169)	41.20 (993)	47.55 (145)
Post-grad		Gov	Private	Self-emp
	Doctorate	46.53 (124)	49.05 (172)	47.22 (79)
	Masters	42.93 (567)	44.42 (863)	47.25 (197)
Some-college		Gov	Private	Self-emp
	Some-college	38.38 (955)	38.73 (5016)	45.70 (704)
University		Gov	Private	Self-emp
	Bachelors	41.56 (943)	42.71 (3455)	46.23 (646)
	Prof-school	48.40 (86)	47.96 (247)	47.78 (209)

Drilling down work

4

Gov		Assoc	Post-grad	Some-college	University
	Federal-gov	41.15 (93)	43.86 (80)	40.31 (251)	43.38 (233)
	Local-gov	41.33 (171)	43.96 (362)	40.14 (385)	42.34 (499)
	State-gov	39.09 (87)	42.93 (249)	34.73 (319)	40.82 (297)
Private		Assoc	Post-grad	Some-college	University
	Private	41.06 (1713)	45.19 (1035)	38.73 (5016)	43.06 (3702)
Self-emp		Assoc	Post-grad	Some-college	University
	Self-emp-inc	48.68 (72)	53.05 (110)	49.31 (223)	49.91 (338)
	Self-emp-not-inc	45.88 (178)	43.39 (166)	44.03 (481)	44.44 (517)

Contributions

- We create a small “movie” that answers an OLAP query
- We complement each query with auxiliary queries organized in thematically related acts that allow us to assess and explain the results of the original query
- We implemented an extensible palette of highlight extraction methods to find interesting patterns in the result of each query
- We describe each highlight with text
- We use TTS technology to convert text to audio

Contributions

- Equally importantly:
  - An extensible software where algorithms for query generation and highlight extraction can be plagued in
  - The demonstration of low technical barrier to produce CineCube reports

Method Overview
Software Issues
Discussion

Method Overview

Our Approach

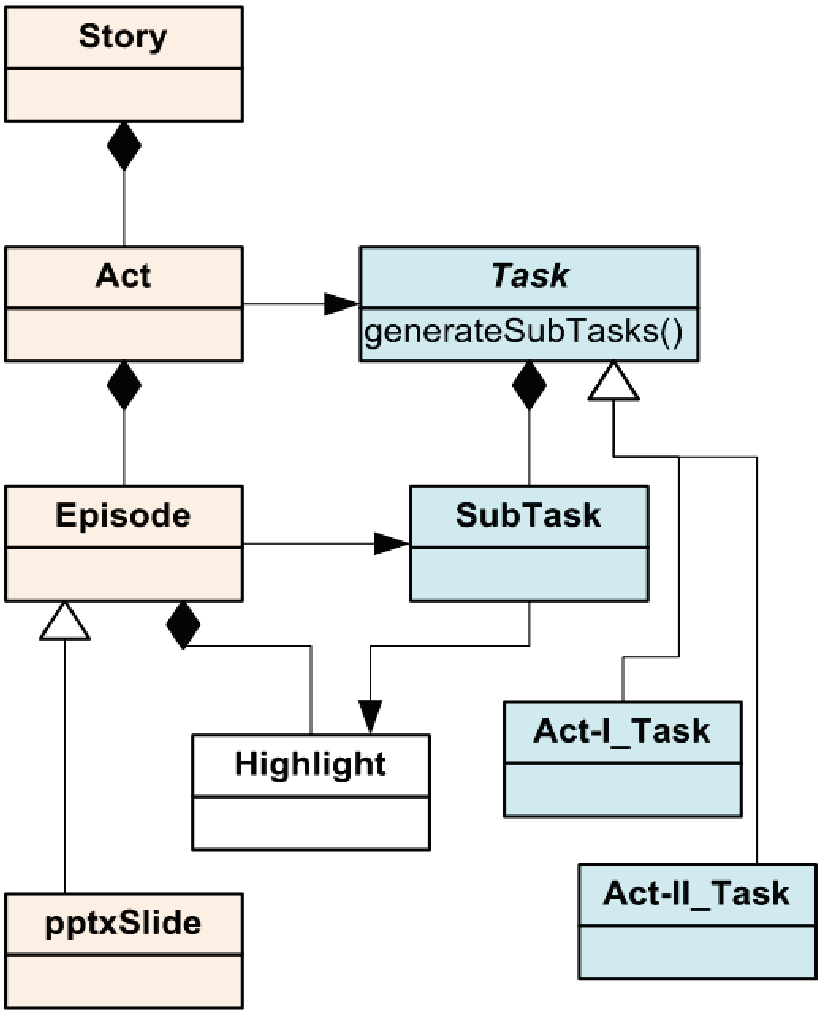
- A first assessment of the current state of affairs
  - Practically, this requirement refers to the execution of the original query.
- Put the state in Context
- Analysis of why things are this way.

Structure of the CineCube Movie

- A typical movie story is structured in acts.
  - Each Act is composed of sequences of scenes
- That's why we organize the CineCube Movie in five Acts:
  - Intro Act
  - Original Act
  - Act I
  - Act II
  - Summary Act

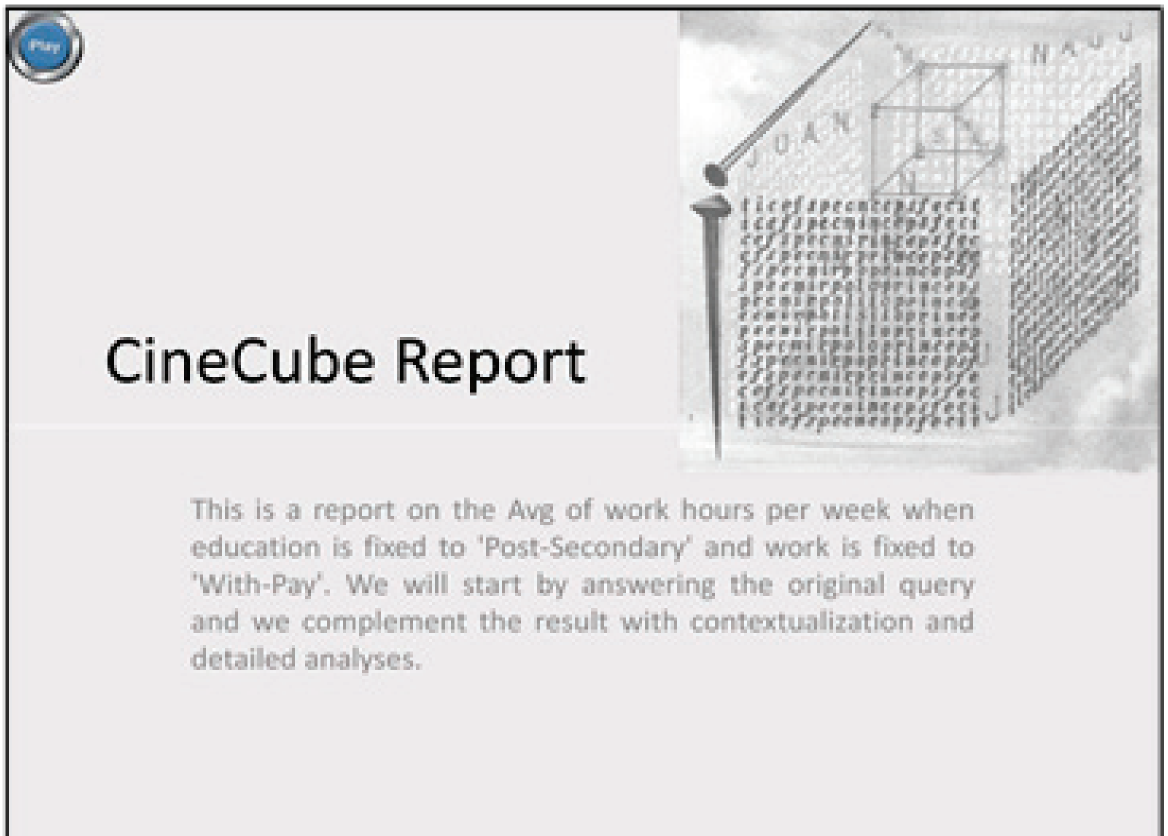
The movie's parts

- Much like movies, we organize our stories in acts
- Each act including several episodes all serving the same purpose



CineCube Movie – Intro Act

- Intro Act has an episode that introduce the story to user



CineCube Movie – Original Act

- Original Act has an episode which is the answer of query that submitted by user

Answer to the original question

	Assoc	Post-grad	Some-college	University
Gov	40.73	43.58	38.38	42.14
Private	41.06	45.19	38.73	43.06
Self-emp	46.68	47.24	45.70	46.61

CineCube Movie – Act I

- In this Act we try to answer the following question:
  - How good is the original query compared to its siblings?
- We compare the marginal aggregate results of the original query to the results of “sibling” queries that use “similar” values in their selection conditions

Act I – Example

Result of Original Query

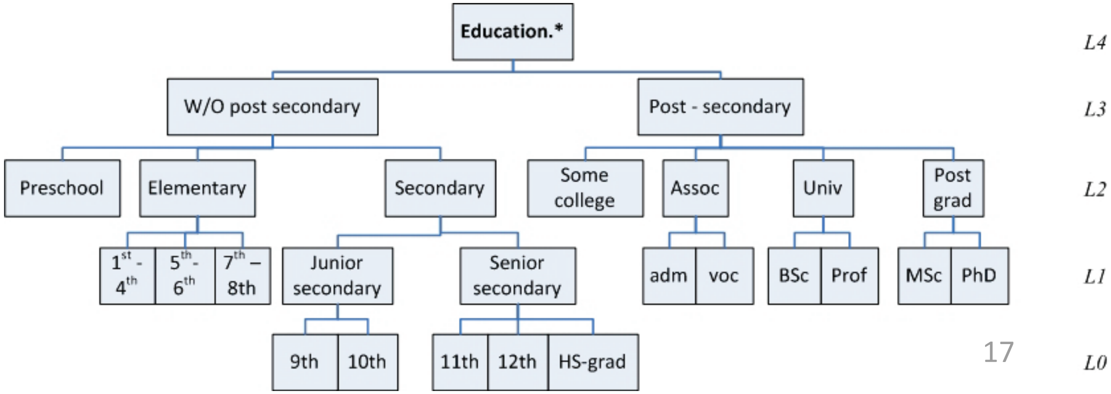
	Assoc	Post-grad	Some-college	University
Gov	40.73	43.58	38.38	42.14
Private	41.06	45.19	38.73	43.06
Self-emp	46.68	47.24	45.70	46.61

$q = (DS^0,$
$W.L_2 = \text{'With-Pay'} \wedge E.L_3 = \text{'Post-Sec'},$
$[W.L_1, E.L_2],$
$\text{avg(Hrs)})$

Assessing the behavior of education

Summary for education	**Post-Secondary**	Without-Post-Secondary
Gov	**41.12**	38.97
Private	**41.06**	39.40
Self-emp	**46.39**	44.84

$q = (DS^0,$
$W.L_2 = \text{'With-Pay'} \wedge E.L_4 = \text{'All'},$
$[W.L_1, E.L_3],$
$\text{avg(Hrs)})$



Act I – Example

Result of Original Query

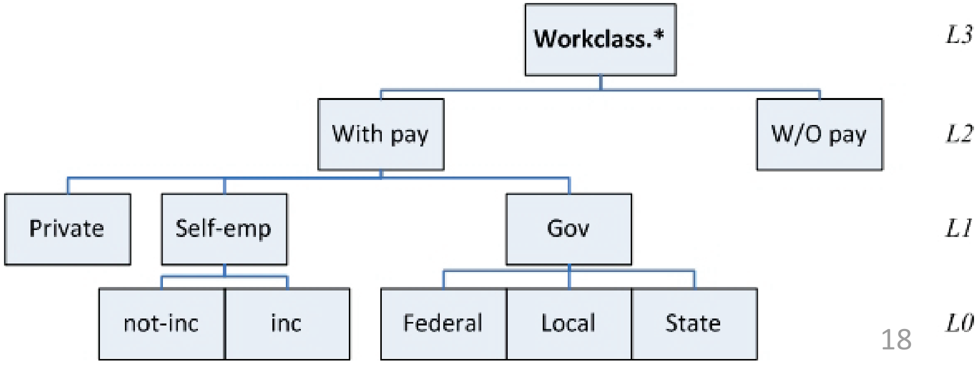
	Assoc	Post-grad	Some-college	University
Gov	40.73	43.58	38.38	42.14
Private	41.06	45.19	38.73	43.06
Self-emp	46.68	47.24	45.70	46.61

q=(DS^0,
W.L_2 ='With–Pay' ∧ E.L_3='Post–Sec',
[W.L_1, E.L_2],
avg(Hrs))

Assessing the behavior of work

Summary for work	Assoc	Post-grad	Some-college	University
With-Pay	**41.62**	**44.91**	**39.41**	**43.44**
Without-pay	50.00	-	35.33	-

q=(DS^0,
W.L_3 ='All' ∧ E.L_3='Post–Sec',
[W.L_2, E.L_2],
avg(Hrs))



CineCube Movie – Act II

- In this Act we try to explaining to user <u>why</u> the result of original query is what it is.
  - “Drilling into the breakdown of the original result”

- We drill in the details of the cells of the original result in order to inspect the internals of the aggregated measures of the original query.

Act II – Example

Result of Original Query

	Assoc	Post-grad	Some-college	University
Gov	40.73	43.58	38.38	42.14
Private	41.06	45.19	38.73	43.06
Self-emp	46.68	47.24	45.70	46.61

$q = (DS^0,$
W.L_2 ='With-Pay' $\wedge$ E.L_3='Post-Sec',
[W.L_1, E.L_2],
avg(Hrs))

Drilling down the Rows of the Original Result

		Assoc	Post-grad	Some-college	University
Gov					
	Federal-gov	41.15 (93)	43.86 (80)	40.31 (251)	43.38 (233)
	Local-gov	41.33 (171)	43.96 (362)	40.14 (385)	42.34 (499)
	State-gov	39.09 (87)	42.93 (249)	34.73 (319)	40.82 (297)
Private					
	Private	41.06 (1713)	45.19 (1035)	38.73 (5016)	43.06 (3702)
Self-emp					
	Self-emp-inc	48.68 (72)	53.05 (110)	49.31 (223)	49.91 (338)
	Self-emp-not-inc	45.88 (178)	43.39 (166)	44.03 (481)	44.44 (517)

Act II – Example

Result of Original Query

	Assoc	Post-grad	Some-college	University
Gov	40.73	43.58	38.38	42.14
Private	41.06	45.19	38.73	43.06
Self-emp	46.68	47.24	45.70	46.61

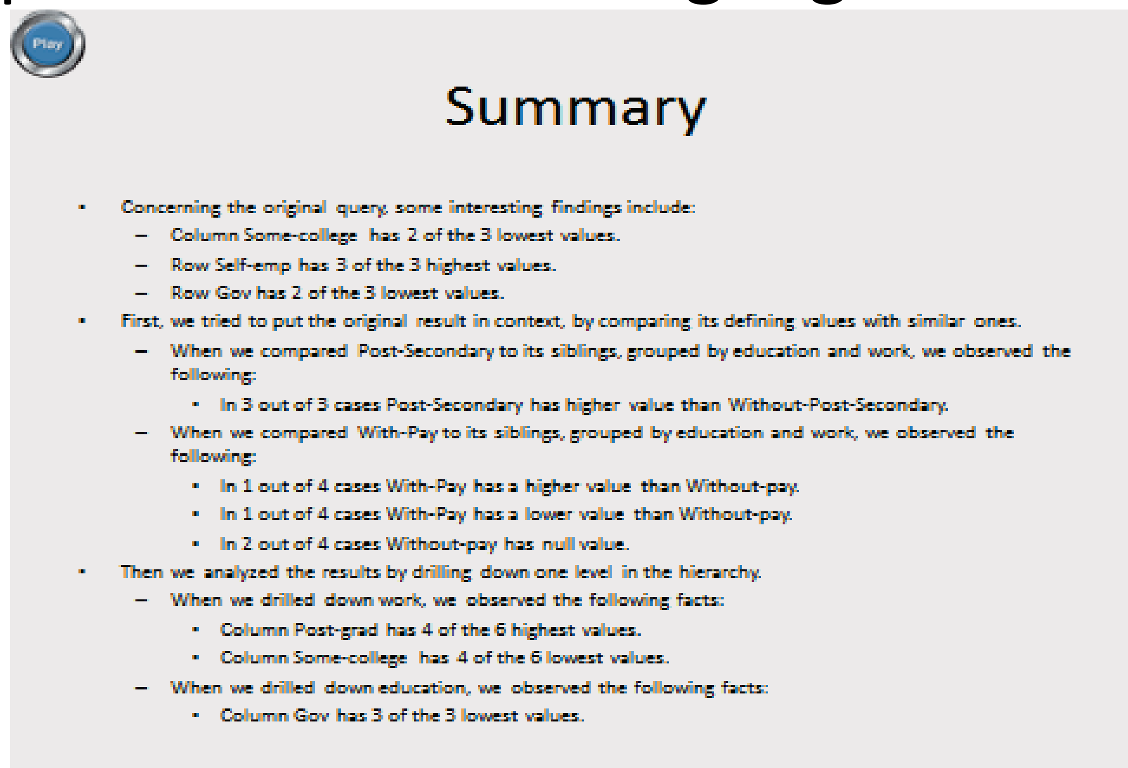
$q = (DS^0,$
$W.L_2 = \text{'With-Pay'} \wedge E.L_3 = \text{'Post-Sec'},$
$[W.L_1, E.L_2],$
$avg(Hrs))$

Drilling down the Columns of the Original Result

		Gov	Private	Self-emp
Assoc				
	Assoc-acdm	39.91 (182)	40.87 (720)	45.49 (105)
	Assoc-voc	41.61 (169)	41.20 (993)	47.55 (145)
Post-grad				
	Doctorate	46.53 (124)	49.05 (172)	47.22 (79)
	Masters	42.93 (567)	44.42 (863)	47.25 (197)
Some-college				
	Some-college	38.38 (955)	38.73 (5016)	45.70 (704)
University				
	Bachelors	41.56 (943)	42.71 (3455)	46.23 (646)
	Prof-school	48.40 (86)	47.96 (247)	47.78 (209)

CineCube Movie – Summary Act

- Summary Act represented from one episode.
- This episode has all the highlights of our story.



Highlight Extraction

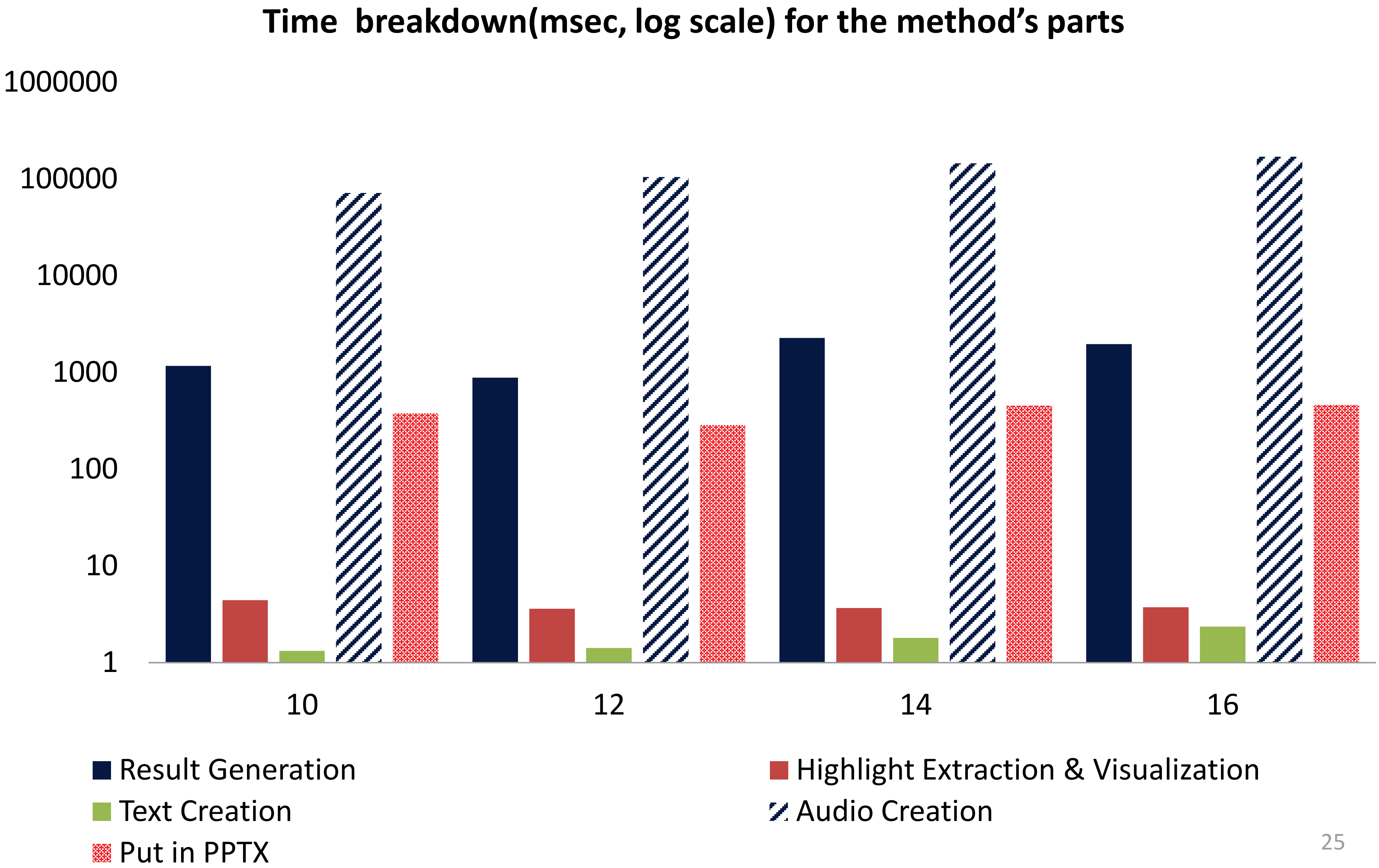
- We utilize a palette of highlight extraction methods that take a 2D matrix as input and produce important findings as output.
- Such as:
  - The top and bottom quartile of values in a matrix
  - The absence of values from a row or column
  - The domination of a quartile by a row or a column
  - The identification of min and max values

Text Extraction

- Text is constructed by a Text Manager that customizes the text per Act
- **Example**:

  In this slide, we drill-down one level for all values of dimension <dim> at level <l>. For each cell we show both the <agg> of <measure> and the number of tuples that correspond to it.

Experimental Results

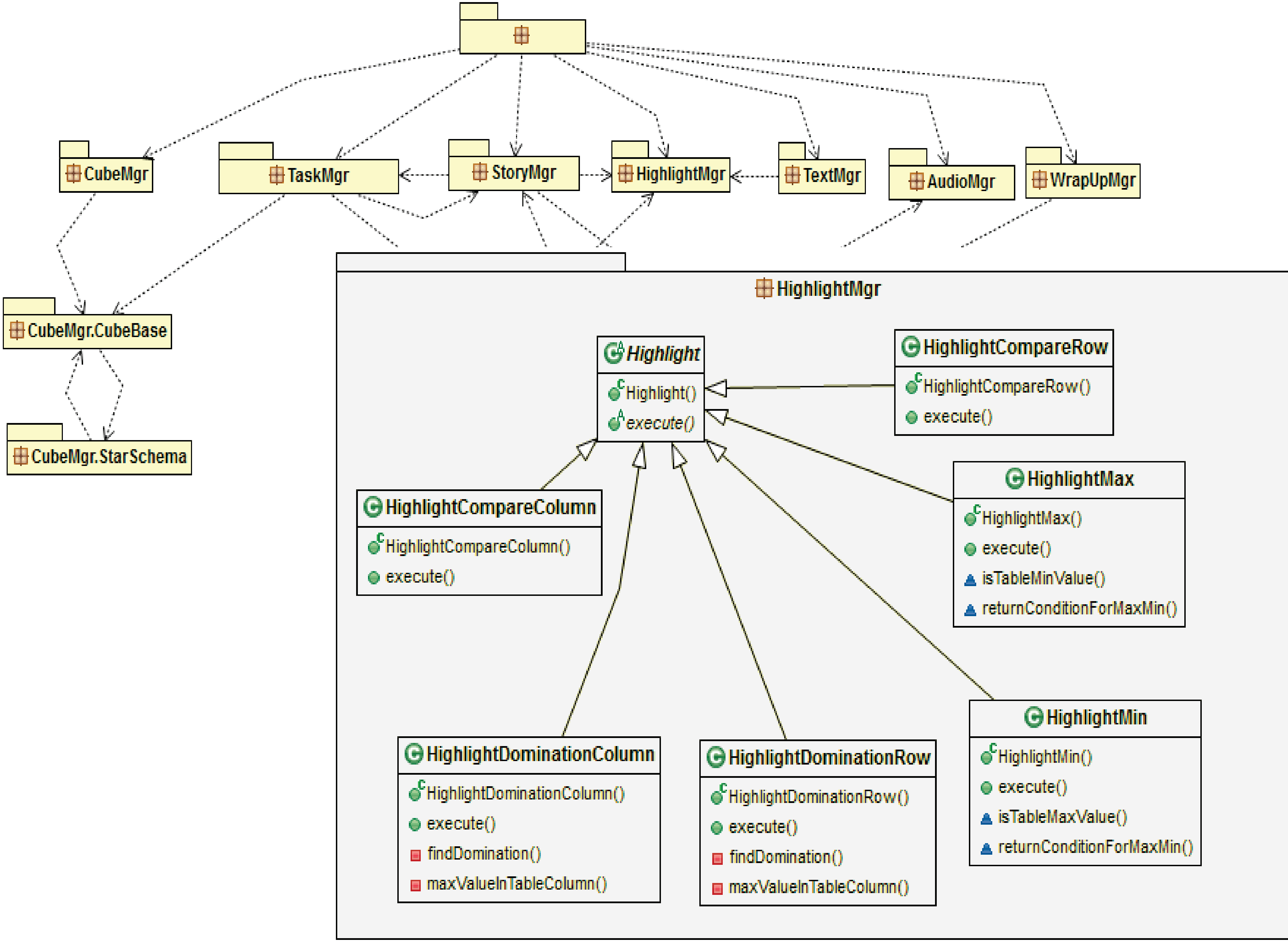


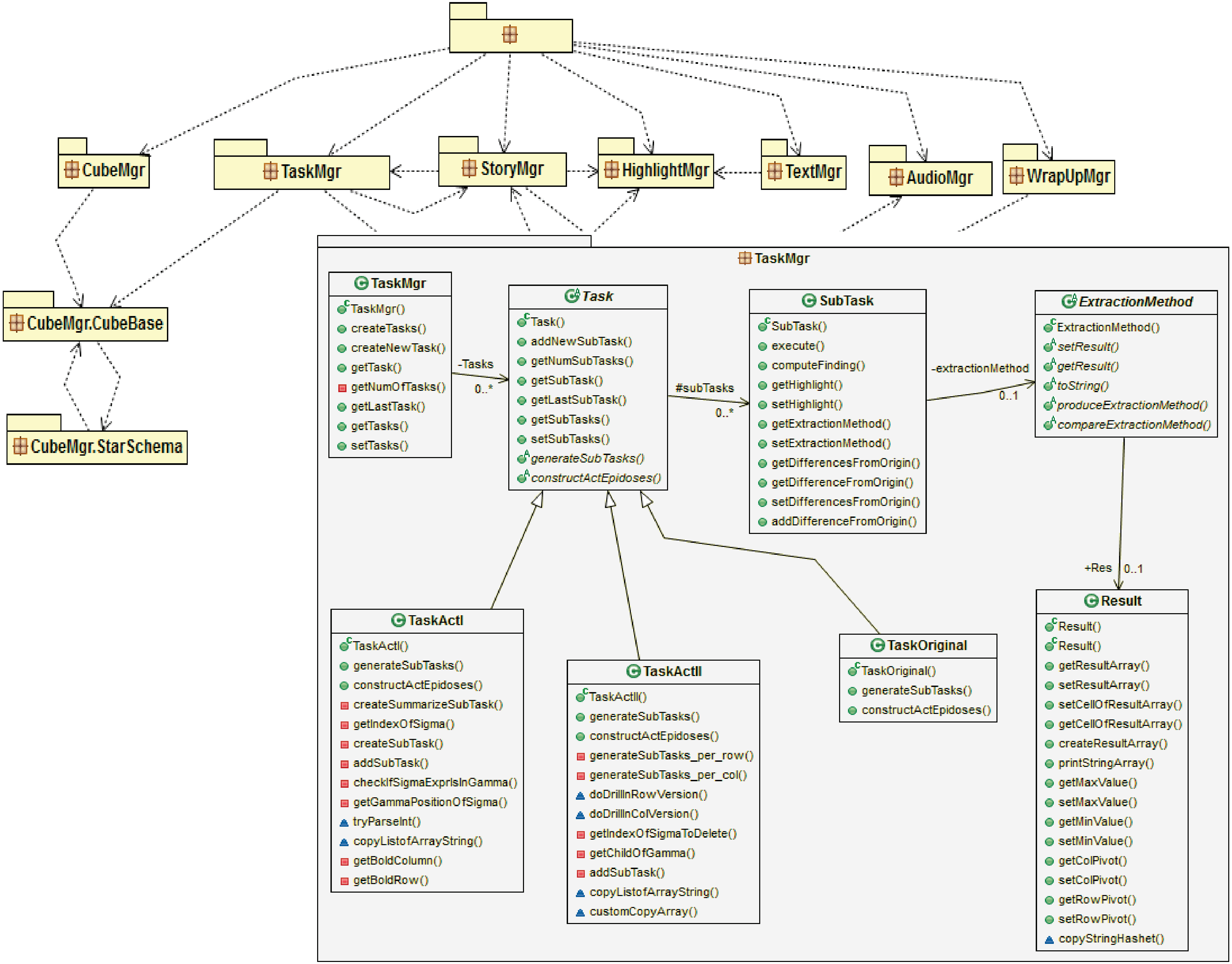
Method Overview
Software Issues
Discussion

Software Issues

Low technical barrier

- Our tool is extensible
  - We can add new tasks to generate complementary queries easily
  - We can add new highlight algorithms to produce highlights easily
- Supportive technologies are surprisingly easier to use
  - Apache POI for pptx generation
  - TTS for text to speech conversion


CubeMgr
TaskMgr
StoryMgr
HighlightMgr
TextMgr
AudioMgr
WrapUpMgr
CubeMgr.CubeBase
CubeMgr.StarSchema
HighlightMgr
Highlight
Highlight()
execute()
HighlightCompareRow
HighlightCompareRow()
execute()
HighlightMax
HighlightMax()
execute()
isTableMinValue()
returnConditionForMaxMin()
HighlightCompareColumn
HighlightCompareColumn()
execute()
HighlightDominationColumn
HighlightDominationColumn()
execute()
findDomination()
maxValueInTableColumn()
HighlightDominationRow
HighlightDominationRow()
execute()
findDomination()
maxValueInTableColumn()
HighlightMin
HighlightMin()
execute()
isTableMaxValue()
returnConditionForMaxMin()


CubeMgr
TaskMgr
StoryMgr
HighlightMgr
TextMgr
AudioMgr
WrapUpMgr
CubeMgr.CubeBase
CubeMgr.StarSchema
TaskMgr
TaskMgr
TaskMgr()
createTasks()
createNewTask()
getTask()
getNumOfTasks()
getLastTask()
getTasks()
setTasks()
-Tasks
0..*
Task
Task()
addNewSubTask()
getNumSubTasks()
getSubTask()
getLastSubTask()
getSubTasks()
setSubTasks()
generateSubTasks()
constructActEpidoses()
#subTasks
0..*
SubTask
SubTask()
execute()
computeFinding()
getHighlight()
setHighlight()
getExtractionMethod()
setExtractionMethod()
getDifferencesFromOrigin()
getDifferenceFromOrigin()
setDifferencesFromOrigin()
addDifferenceFromOrigin()
-extractionMethod
0..1
ExtractionMethod
ExtractionMethod()
setResult()
getResult()
toString()
produceExtractionMethod()
compareExtractionMethod()
+Res
0..1
Result
Result()
Result()
getResultArray()
setResultArray()
setCellOfResultArray()
getCellOfResultArray()
createResultArray()
printStringArray()
getMaxValue()
setMaxValue()
getMinValue()
setMinValue()
getColPivot()
setColPivot()
getRowPivot()
setRowPivot()
copyStringHashet()
TaskActI
TaskActI()
generateSubTasks()
constructActEpidoses()
createSummarizeSubTask()
getIndexOfSigma()
createSubTask()
addSubTask()
checkIfSigmaExprIsInGamma()
getGammaPositionOfSigma()
tryParseInt()
copyListofArrayString()
getBoldColumn()
getBoldRow()
TaskActII
TaskActII()
generateSubTasks()
constructActEpidoses()
generateSubTasks_per_row()
generateSubTasks_per_col()
doDrillInRowVersion()
doDrillInColVersion()
getIndexOfSigmaToDelete()
getChildOfGamma()
addSubTask()
copyListofArrayString()
customCopyArray()
TaskOriginal
TaskOriginal()
generateSubTasks()
constructActEpidoses()

Apache POI for pptx

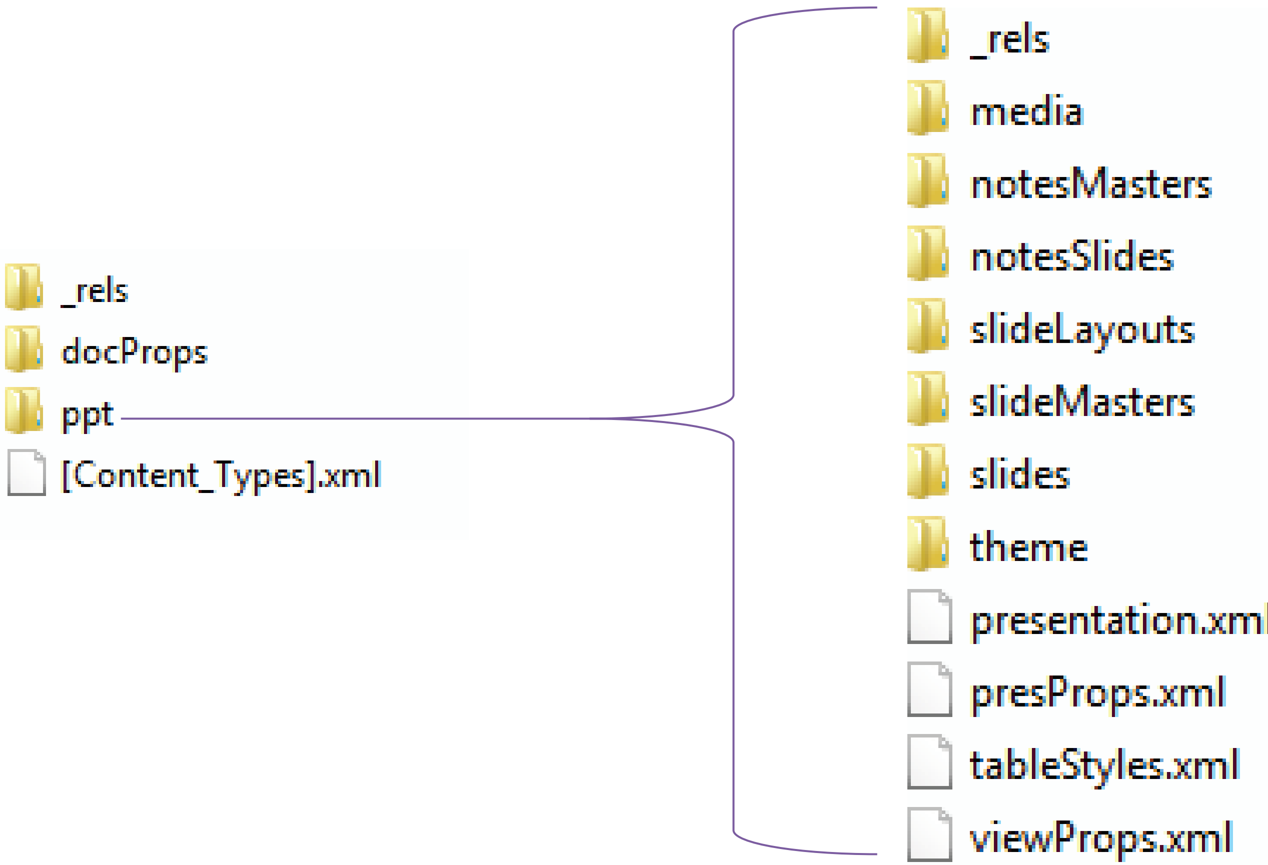
- A Java API that provides several libraries for Microsoft Word, PowerPoint and Excel (since 2001).
- XSLF is the Java implementation of the PowerPoint 2007 OOXML (.pptx) file format.

```
XMLSlideShow ss = new XMLSlideShow();
XSLFSlideMaster sm = ss.getSlideMasters()[0];

XSLFSlide sl= ss.createSlide
(sm.getLayout(SlideLayout.TITLE_AND_CONTENT));

XSLFTable t = sl.createTable();
t.addRow().addCell().setText(“added a cell”);
```

PPTX Folder Structure



MaryTTS for Text-to-Speech Synthesis

```
MaryInterface m = new LocalMaryInterface();
m.setVoice(“cmu-slt-hsmm”);

AudioInputStream audio = m.generateAudio("Hello”);

AudioSystem.write(audio, audioFileFormat.Type.WAVE,
new File(“myWav.wav”));
```

Method Overview
Software Issues
Discussion

Discussion

Open issues

Be compendious; if not, at least be concise!

Visualize

Show text

More than 2D arrays

Look like a movie

Back stage

Multi-query

Speed-up voice gen.

Cloud/parallel

Info content

Structure more like a movie

More highlights

More acts (more queries)

Crowd wisdom

Chase after interestingness

CC now

Assumptions

Star schema

2D results (2 groupers)

Single measure

Equality selections

Interaction

How to allow interaction with the user?

Personalization

Thank you!

Any questions?

More information

- http://www.cs.uoi.gr/~pvassil/projects/cinecubes/

Demo

- http://snf-56304.vm.okeanos.grnet.gr/

Code

- https://github.com/DAINTINESS-Group/CinecubesPublic.git

AUXILIARY SLIDES

Related Work

Related Work

- Query Recommendations
- Database-related efforts
- OLAP-related methods
- Advanced OLAP operators
- Text synthesis from query results

Query Recommendations

- A. Giacometti, P. Marcel, E. Negre, A. Soulet, 2011. Query Recommendations for OLAP Discovery-Driven Analysis. IJDWM 7,2 (2011), 1-25 DOI= http://dx.doi.org/10.4018/jdwm.2011040101
- C. S. Jensen, T. B. Pedersen, C. Thomsen, 2010. Multidimensional Databases and Data Warehousing. Synthesis Lectures on Data Management, Morgan & Claypool Publishers
- A. Maniatis, P. Vassiliadis, S. Skiadopoulos, Y. Vassiliou, G. Mavrogonatos, I. Michalarias, 2005. A presentation model and non-traditional visualization for OLAP. IJDWM, 1,1 (2005), 1-36. DOI= http://dx.doi.org/10.4018/jdwm.2005010101
- P. Marcel, E. Negre, 2011. A survey of query recommendation techniques for data warehouse exploration. EDA (Clermont-Ferrand, France, 2011), pp. 119-134

Database-related efforts

- K. Stefanidis, M. Drosou, E. Pitoura, 2009. "You May Also Like" Results in Relational Databases. PersDB (Lyon, France, 2009).

- G. Chatzopoulou, M. Eirinaki, S. Koshy, S. Mittal, N. Polyzotis, J. Varman, 2011. The QueRIE system for Personalized Query Recommendations. IEEE Data Eng. Bull. 34,2 (2011), pp. 55-60

OLAP-related methods

- V. Cariou, J. Cubillé, C. Derquenne, S. Goutier, F.Guisnel, H. Klajnmic, 2008. Built-In Indicators to Discover Interesting Drill Paths in a Cube. DaWaK (Turin, Italy, 2008), pp. 33-44, DOI=http://dx.doi.org/10.1007/978-3-540-85836-2_4

- A. Giacometti, P. Marcel, E. Negre, A. Soulet, 2011. Query Recommendations for OLAP Discovery-Driven Analysis. IJDWM 7,2 (2011), 1-25 DOI= http://dx.doi.org/10.4018/jdwm.2011040101

Advanced OLAP operators

- Sunita Sarawagi: User-Adaptive Exploration of Multidimensional Data. VLDB 2000:307-316
- S. Sarawagi, 1999. Explaining Differences in Multidimensional Aggregates. VLDB (Edinburgh, Scotland, 1999), pp. 42-53
- G. Sathe, S. Sarawagi, 2001. Intelligent Rollups in Multidimensional OLAP Data. VLDB (Roma, Italy 2001), pp.531-540

Text synthesis from query results

- A. Simitsis, G. Koutrika, Y. Alexandrakis, Y.E. Ioannidis, 2008. Synthesizing structured text from logical database subsets. EDBT (Nantes, France, 2008) pp. 428-439, DOI=http://doi.acm.org/10.1145/ 1353343.1353396

Formalities

OLAP Model

- We base our approach on an OLAP model that involves
  - Dimensions, defined as lattices of dimension levels
  - Ancestor functions, (in the form of $\mathrm{anc}_{L_1}^{L_2}$) mapping values between related levels of a dimension
  - Detailed data sets, practically modeling fact tables at the lowest granule of information
  - Cubes, defined as aggregations over detailed data sets

What is Cube?

- A primary Cube C is described as

$$C=(DS^0, \phi, [L_1, \dots, L_n, M_1, \dots, M_m], [agg_1(M_1^0), \dots, agg_1(M_m^0)])$$

  - DS^0 is a detailed dataset over the schema
  - Φ is a detailed selection condition
    - Φ analyzed as $\varphi_1 \wedge \cdots \wedge \varphi_k$
    - $\varphi_i \ is \ D_i.L_j = value_i$
  - $L_1, \dots, L_n$ are levels such that $L_i \prec L_{i+1}$, $1 \leq i \leq n$.
  - $M_1, \dots, M_m$ are measures
  - $agg_i \in \{max, min, sum, count, \text{average}\}$, $1 \leq i \leq m$

Cube Query

- A cube query Q can be considered as

$$Q=(DS^0,\Sigma,\Gamma,\gamma(M))$$

- where:
  - Σ is a conjunction of dimensional restrictions of the form
  - Γ is a set of grouper dimensional level
  - γ(M) is an aggregate function applied to the measure of the cube

Cube Query

- In our approach we assume that the user submit cube queries which denote as:
    - $q=(DS^0, \varphi_1 \land \cdots \land \varphi_k, [L_\alpha, L_\beta], agg(M))$
- Example:

$q=(A, W.L_2 = \text{'With-Pay'} \land E.L_3 = \text{'Post-Sec'}, [W.L_1, E.L_2], avg(Hrs))$

Cube Query to SQL Query

- In general case :

  SELECT $L_1, \ldots, L_n, agg_1(M_1^0), \ldots, agg_1(M_1^0)])$
  FROM DS^0 INNER JOIN D_1, ...INNER JOIN D_n
  WHERE ϕ
  GROUP BY $L_1, \ldots, L_n$

- Example for our case:

  SELECT W.L_1, E.L_2,AVG(Hrs)
  FROM A
  INNER JOIN W ON A.W=W.L_0
  INNER JOIN E ON A.E=E.L_0
  WHERE W.L_2 ='With-Pay' AND E.L_3 ='Post-Sec'
  GROUP BY W.L_1, E.L_2

Method Internals

Act I – Our Definition

- We introduce two marginal sibling queries, one for each aggregator.
- Formally, given an original query:

$$q = (DS^0,\ \varphi_1 \wedge \cdots \wedge \varphi_k, [L_\alpha,\ L_\beta],\ agg(M))$$

- Its two marginal sibling queries are:
  1. $q^s = (DS^0,\ \varphi_1 \wedge \cdots \varphi^*_\chi \wedge \cdots \wedge \varphi_k, [L_\alpha,\ L_\chi],\ agg(M))$
  2. $q^s = (DS^0,\ \varphi_1 \wedge \cdots \varphi^*_\chi \wedge \cdots \wedge \varphi_k, [L_\chi,\ L_\beta],\ agg(M))$
     - $\varphi^*_\chi : L_{x+1} = anc_{L_x}^{L_{x+1}}(v)$

Act I – Query Example

- Original Query
  - q=(DS^0,W.L_2 ='With–Pay' $\wedge$ E.L_3='Post–Sec', [W.L_1, E.L_2], avg(Hrs))
- Sibling Queries:
  1. q=(DS^0,W.L_2='With–Pay' $\wedge$ E.L_4='All', [W.L_1, E.L_3], avg(Hrs))
  2. q=(DS^0,W.L_3 ='All' $\wedge$ E.L_3='Post–Sec', [W.L_2, E.L_2], avg(Hrs))

Act I – How produce it?

- We define a sibling query as a query with a single difference to the original:
  - Instead of an atomic selection formula $L_i = v_i$, the sibling query contains a formula of the form $L_i \in$ childen(parent(v_i)).
- Formally, given an original query

$$q = (DS^0, \varphi_1 \wedge \cdots \wedge \varphi_k, [L_\alpha, L_\beta], agg(M))$$

- A new query q^s is a sibling query if is of the form

$$q^s = (DS^0, \varphi_1 \wedge \cdots \varphi_\chi^* \wedge \cdots \wedge \varphi_k, [L_\alpha, L_\beta], agg(M))$$

  - $\varphi_\chi^*: L_{x+1} = anc_{L_x}^{L_{x+1}}(v)$

Act II – Query Example

- Original Query
  - $q=(DS^0, W.L_2 = \text{'With-Pay'} \wedge E.L_3 = \text{'Post-Sec'}, [W.L_1, E.L_2], avg(Hrs))$
- Drill in Queries for work dimension:
  1. $q=(DS^0, W.L_1 = \text{'Gov'} \wedge E.L_3 = \text{'Post-Sec'}, [W.L_0, E.L_2], avg(Hrs))$
  2. $q=(DS^0, W.L_1 = \text{'Private'} \wedge E.L_3 = \text{'Post-Sec'}, [W.L_0, E.L_2], avg(Hrs))$
  3. $q=(DS^0, W.L_1 = \text{'Self-emp'} \wedge E.L_3 = \text{'Post-Sec'}, [W.L_0, E.L_2], avg(Hrs))$

For Education dimension: similarly

Act II- How produce it?

- Assume a cube query and its result, visualized as a 2D matrix.
- For each cell c of this result is characterized by the following cube query:
  - $q^c = (DS^0, \phi_1 \wedge \ldots \wedge \phi_k \wedge \phi_c, [L_\alpha, L_\beta], agg(M))$
    - $\varphi_c : L_\alpha = v_a^c \wedge L_\beta = v_\beta^c$

Act II- How produce it?

- For each of the aggregator dimensions, we can generate a set of explanatory drill in queries, one per value in the original result:

1. q_a^s=(DS^0, $\phi_1 \wedge \ldots \wedge \phi_k \wedge \phi$,[$L_{\alpha-1}$,$L_\beta$],agg(M)),
2. q_β^s=(DS^0, $\varphi_1 \wedge \cdots \wedge \varphi_k \wedge \varphi_c$,[$L_\alpha$, $L_{\beta-1}$], agg(M))
   - φ_c : $L_\alpha = v_a^c \wedge L_\beta = v_\beta^c$

Our Algorithm

Algorithm Construct Operational Act

Input: the original query over the appropriate database

Output: a set of an act's episodes fully computed

1. Create the necessary objects (act, episodes, tasks, subtasks) appropriately linked to each other
2. Construct the necessary queries for all the subtasks of the Act, execute them, and organize the result as a set of aggregated cells (each including its coordinates, its measure and the number of its generating detailed tuples)
3. For each episode

– Calculate the visual presentation of cells

– Calculate the cells' highlights

– Produce the text based on the highlights

– Produce the audio based on the text

Experiments and Results

Experimental setup

- Adult dataset referring to data from 1994 USA census
  - Has 7 dimension Age, Native Country, Education, Occupation, Marital status, Work class, and Race.
  - One Measure : work hours per week
- Machine Setup :
  - Running Windows 7
  - Intel Core Duo CPU at 2.50GHz,
  - 3GB main memory.

Experiments