CT-OLAP: Temporal Multidimensional Model and Algebra for Moving Objects

Eilverijus Kondratas and Igor Timko

Free University of Bozen-Bolzano, Italy

DOLAP 2007,November 9, 2007, Lisbon, Portugal

Outline

- Problem
- Current Technology
- Our approach
  - Model Extension
  - Algebra Extension
- Implementation
- Conclusions and Future Work

Outline

- Problem
- Current Technology
- Our approach
  - Model Extension
  - Algebra Extension
- Implementation
- Conclusions and Future Work

Problem - Traffic Jams

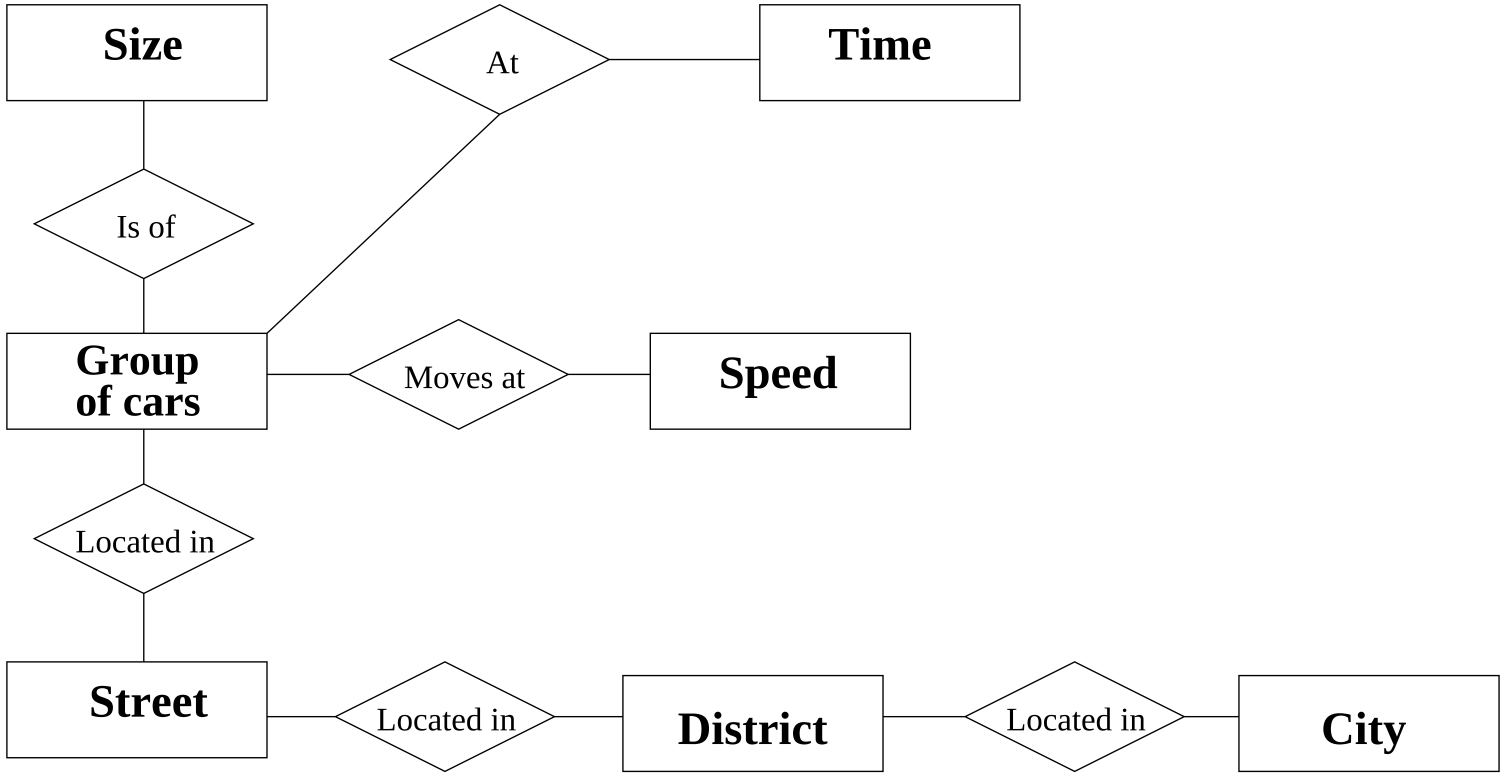
- Traffic jams is nowadays problem, especially in large cities
- Governments put a lot of effort to reduce traffic jams

Problem - Traffic Jams

- Traffic jams is nowadays problem, especially in large cities
- Governments put a lot of effort to reduce traffic jams
- Traffic jams analysts need tools to help in modeling and analyzing data
- Ongoing project "Bolzano in 10 minutes" (BZ10M) with municipality of Bolzano

Objects of Interest

- We are interested in car groups



Query: "How the number of cars was changing per each district in the last three months?"

Outline

- Problem
- Current Technology
- Our approach
  - Model Extension
  - Algebra Extension
- Implementation
- Conclusions and Future Work

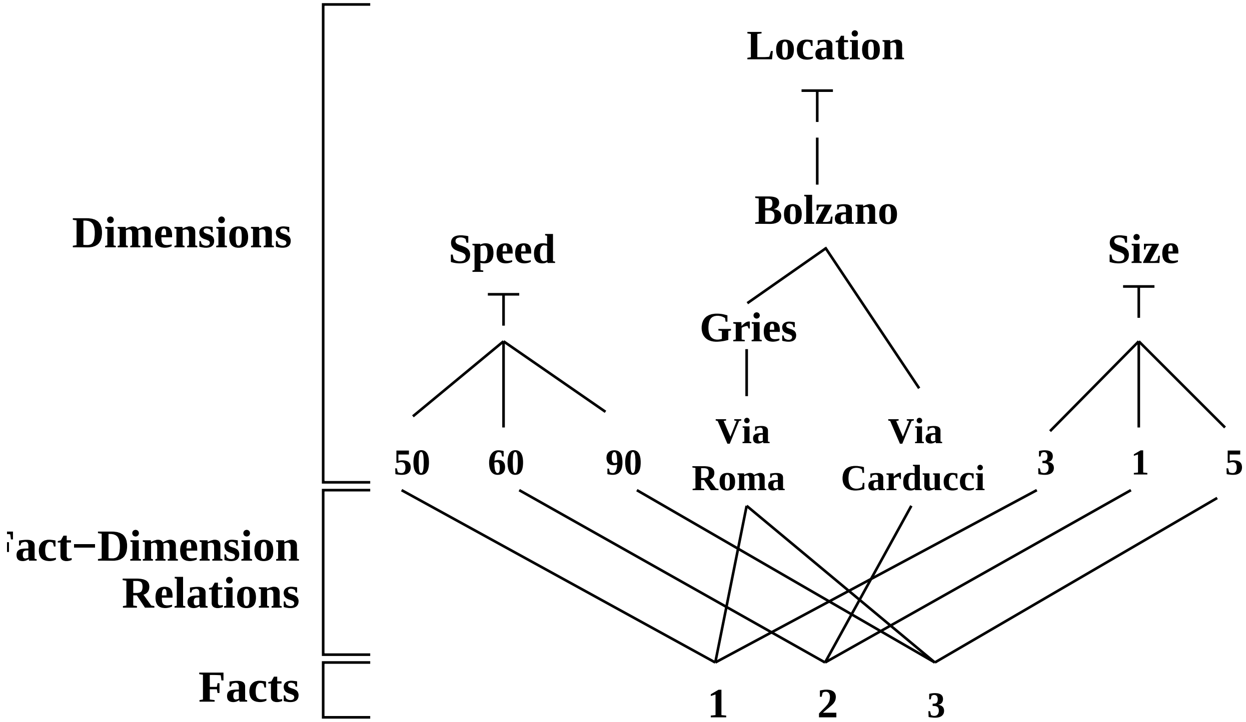
Current Technology

- Temporal On-Line Analytical processing (TOLAP)[1,2,3] is a technology to answer temporal analytical queries
- Industrial and research fields support only discrete time TOLAP
- A problem:
  - A car (groups of cars) which moves continuously can not be directly represented in TOLAP

Current Technology

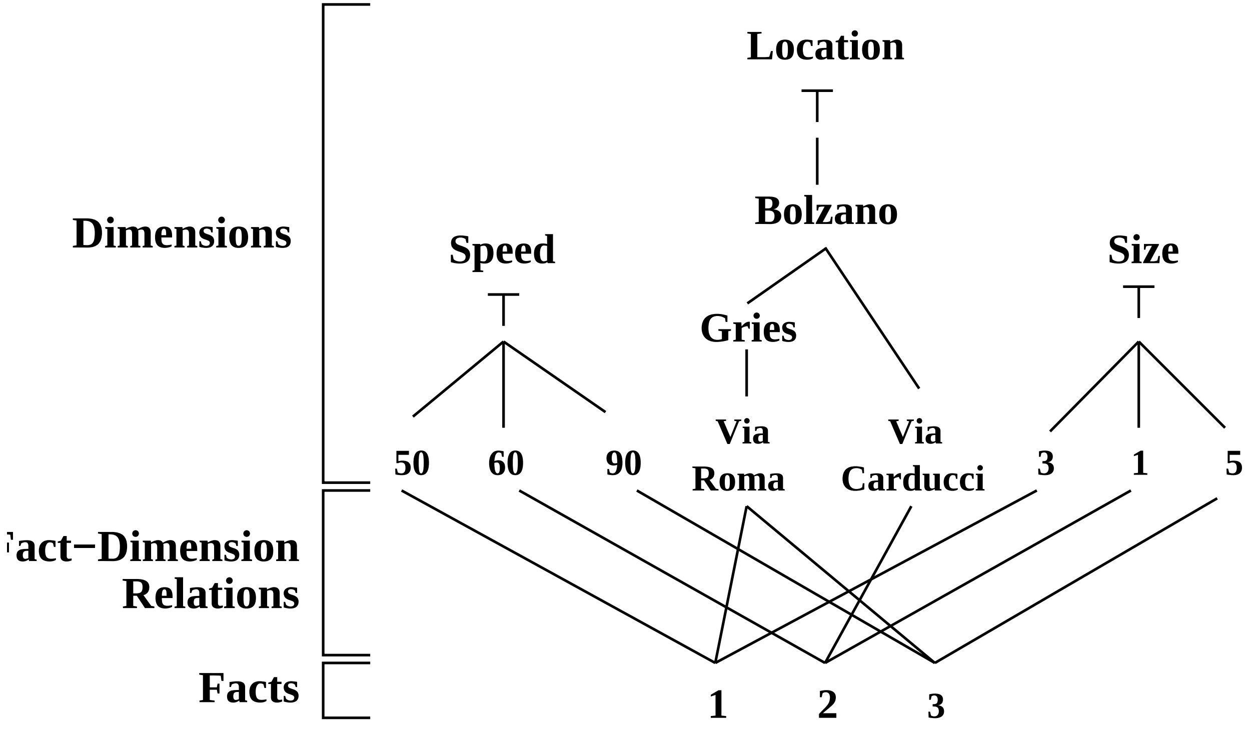
- Temporal On-Line Analytical processing (TOLAP)[1,2,3] is a technology to answer temporal analytical queries
- Industrial and research fields support only discrete time TOLAP
- A problem:
  - A car (groups of cars) which moves continuously can not be directly represented in TOLAP
- We want to extend existing TOLAP model and algebra to handle continuous time

Current Model



- We use model defined by Pedersen et al.: "A Foundation for Capturing and Querying Complex Multidimensional Data", *Information Systems*, 26(5):383–423, 2001.

Current Model

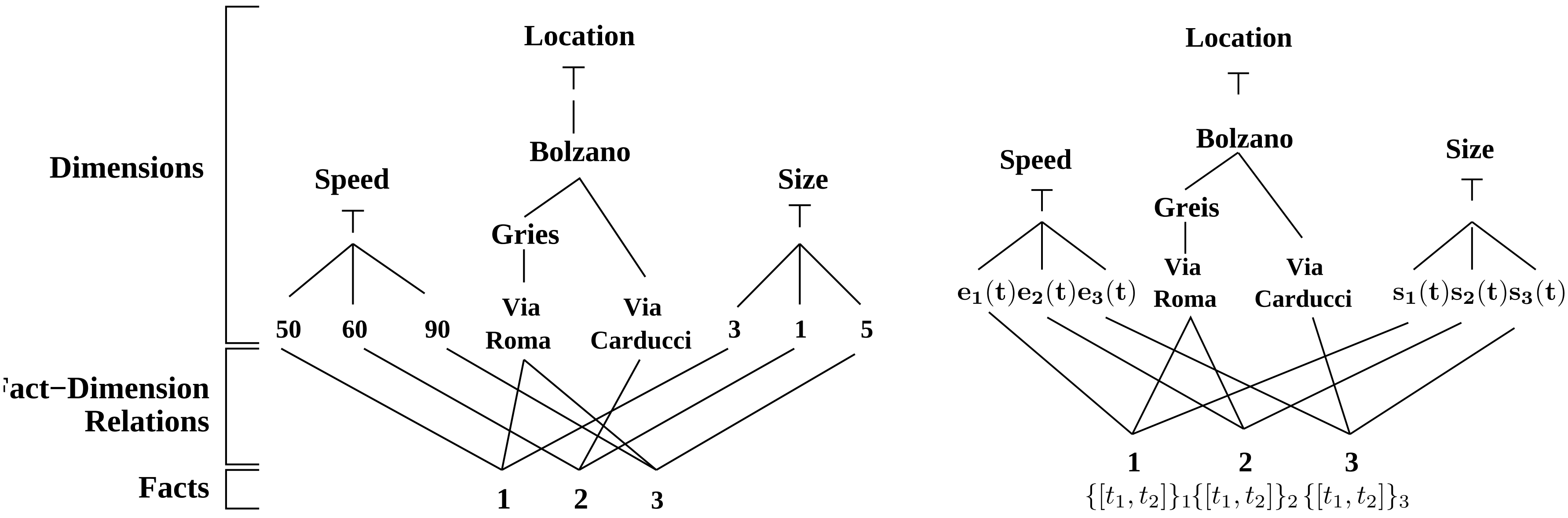


- Fact are objects of interest
- Dimensions are properties of facts
- Fact-Dimension relations relate facts to particular dimension values

Outline

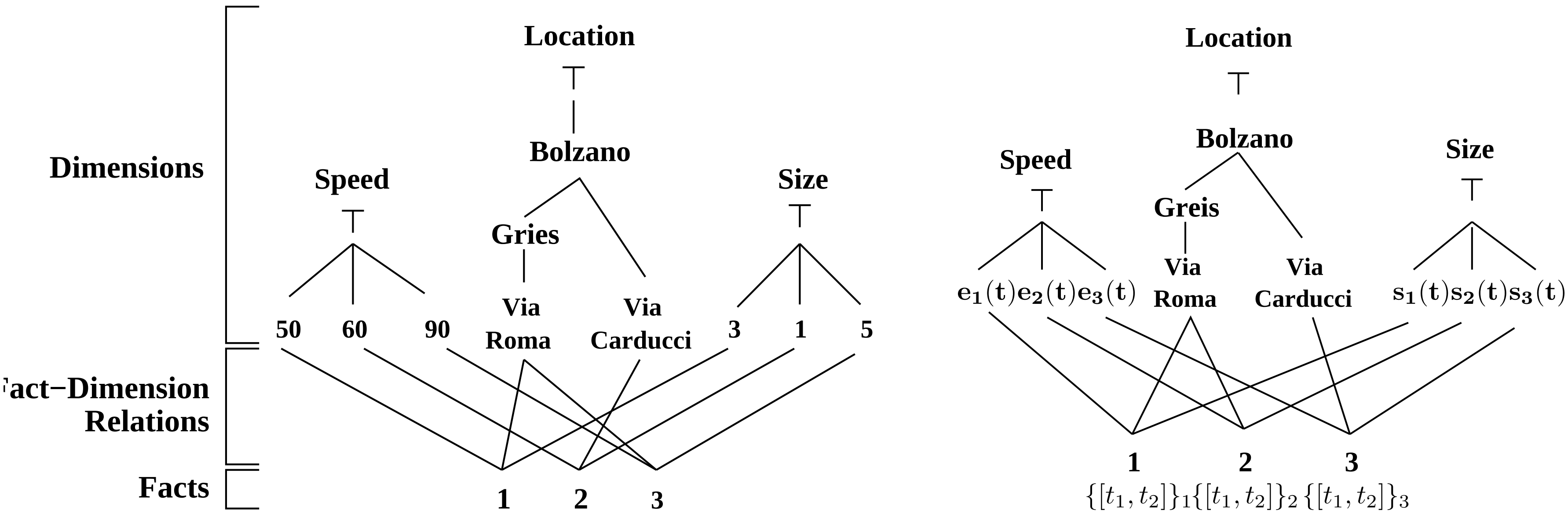
- Problem
- Current Technology
- Our approach
  - Model Extension
  - Algebra Extension
- Implementation
- Conclusions and Future Work

Extended Model



- We substitute single dimension values with *dimension functions*
- Fact valid time is associated with a fact itself

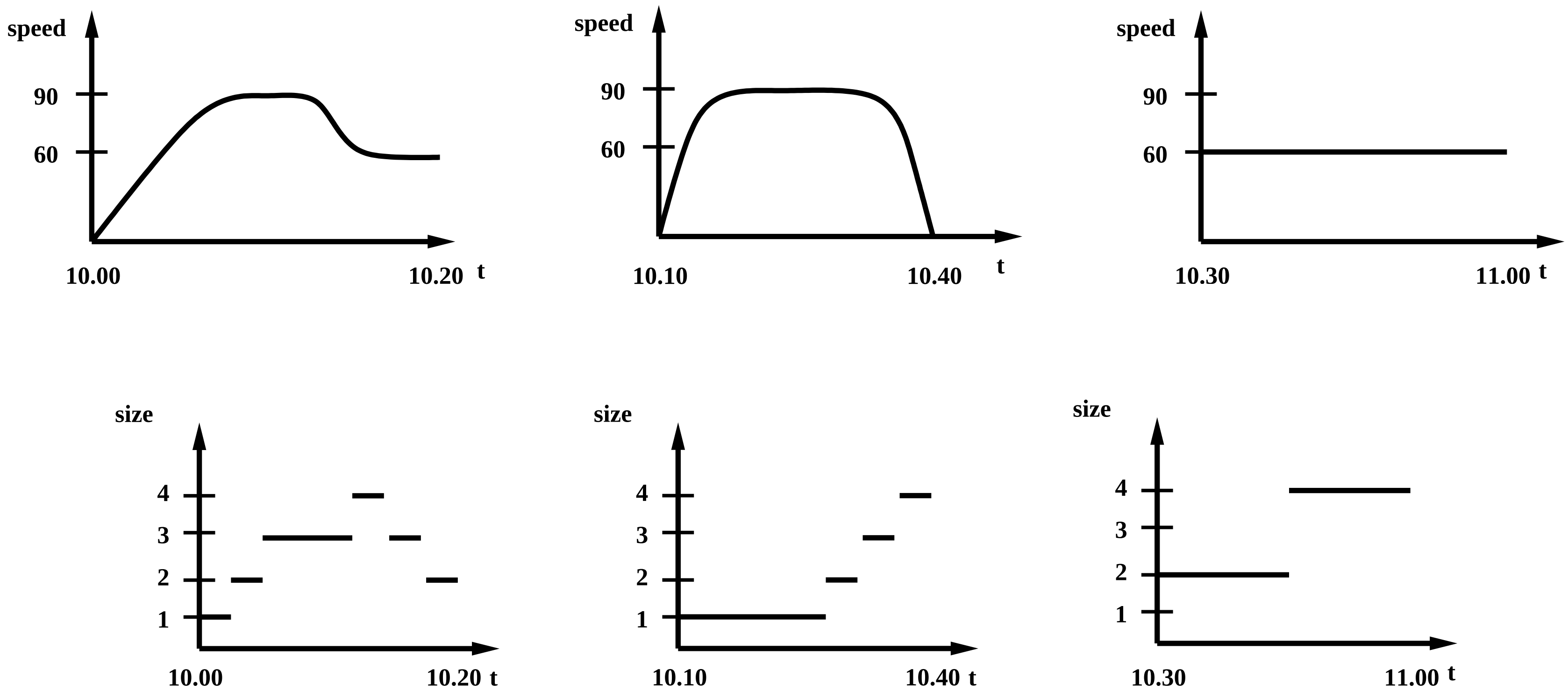
Extended Model



- Two dimension types: *basic* and *extended*
  - *Basic* type dimensions hold only constant functions
  - *Extended* type dimensions hold continuous, constant step-wise, etc. functions

Dimension Functions

- Dimension functions

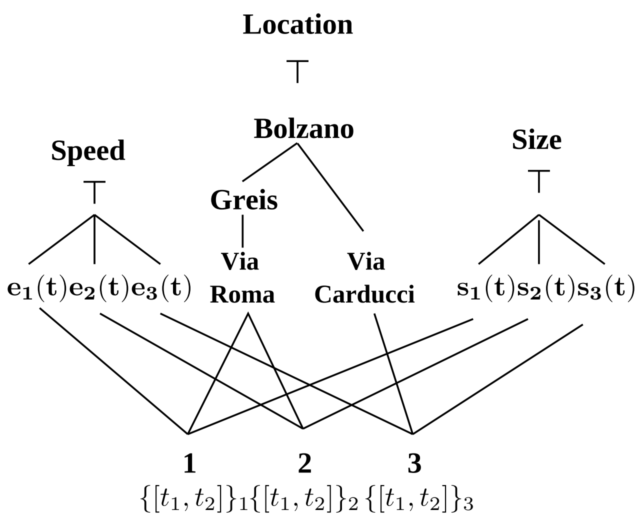


Algebra

- We want to make algebra at least as powerful as BCDM [4] excluding transaction time
- Algebra operators
  - Select
  - Project (borrowed from Pedersen et al.)
  - Union (borrowed from Pedersen et al.)
  - Difference
  - Identity-based join
  - Aggregate formation

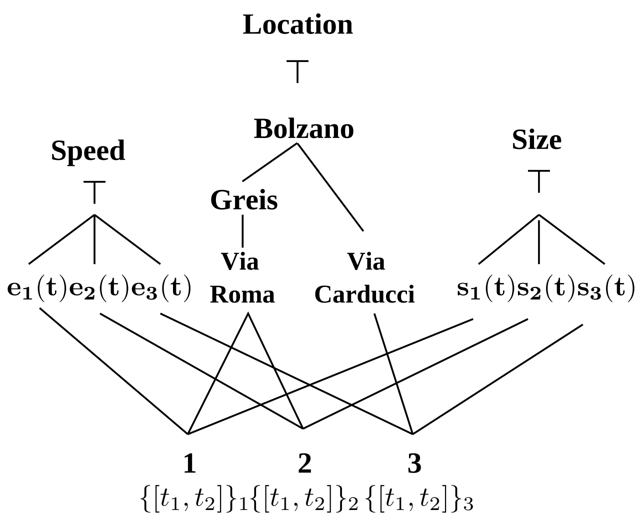
Selection (1/2)

- We perform sequenced (for every instance of time) select that restricts set of facts to those that satisfy given predicate
- Select groups of cars of size 3 or bigger that were moving less than 30 km/h ($p = (speed < 30 \wedge size \geq 3)$)



Selection (1/2)

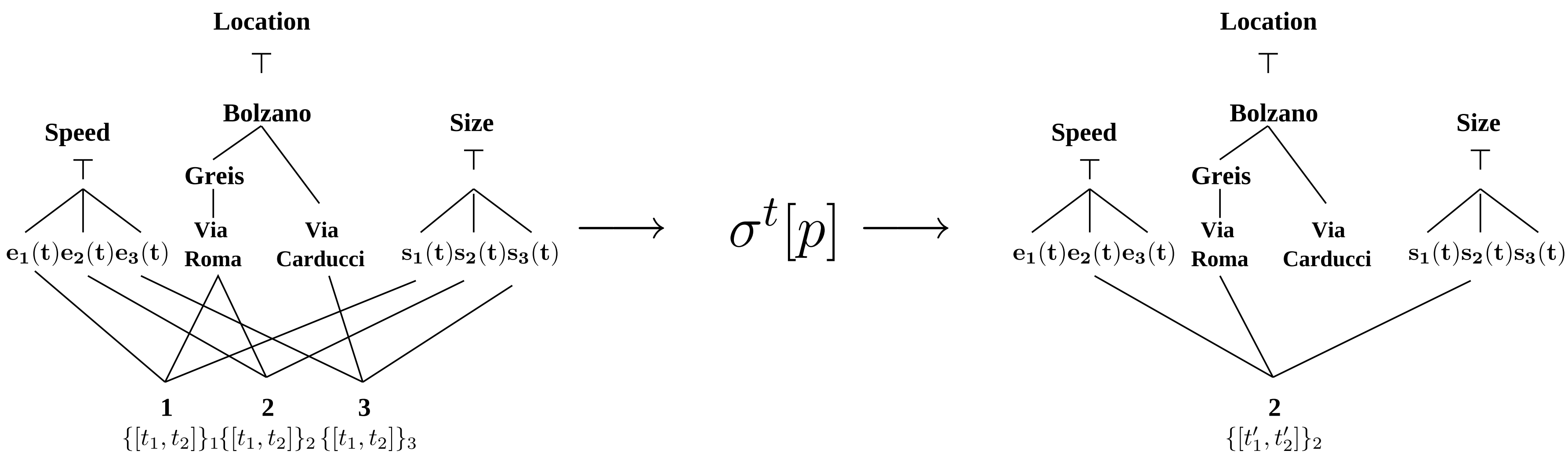
- We perform sequenced (for every instance of time) select that restricts set of facts to those that satisfy given predicate
- Select groups of cars of size 3 or bigger that were moving less than 30 km/h ($p = (speed < 30 \wedge size \geq 3)$)



$\longrightarrow \quad \sigma^t[p] \quad \longrightarrow$

Selection (1/2)

- We perform sequenced (for every instance of time) select that restricts set of facts to those that satisfy given predicate
- Select groups of cars of size 3 or bigger that were moving less than 30 km/h ($p = (speed < 30 \wedge size \geq 3)$)



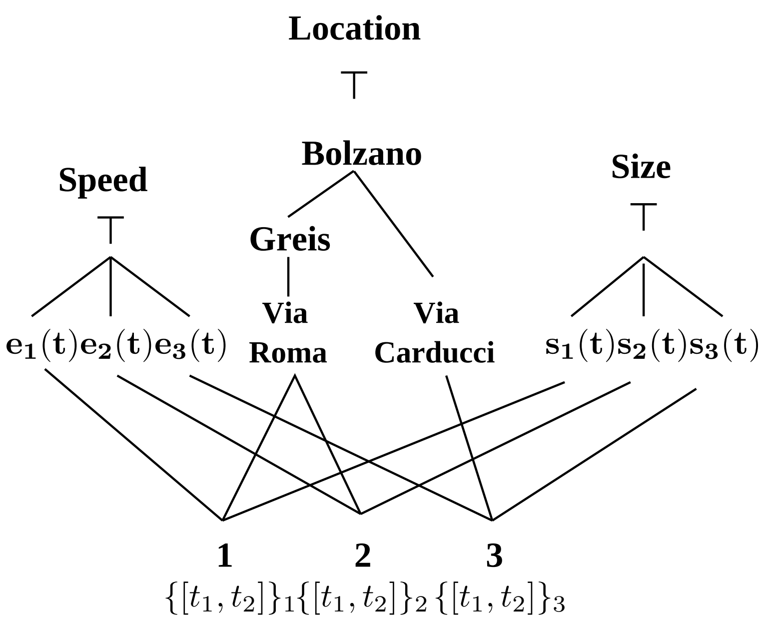
Selection (2/2)

Given a n-dimensional $MO = (\mathcal{S}, F, D, R)$ and predicate p on the dimensions $D = \{D_1, \ldots, D_n\}$, we define the *CT selection* operator, σ^t, as follows: $\sigma^t[p](M) = (\mathcal{S}, F', D, R')$ where

$$
\begin{aligned}
F' = \{(f, T'_f) | (f, T_f) \in F \wedge & \\
& (\exists T'_f \subseteq T_f (\forall t \in T'_f \\
& (\exists e_1 \in D_1, ..., e_n \in D_n | p(e_1(t), ..., e_n(t)))))\}, \\
R' = R'_i, R'_i = \{(f, e(t)) | (f, T'_f) \in F'\}.
\end{aligned}
$$

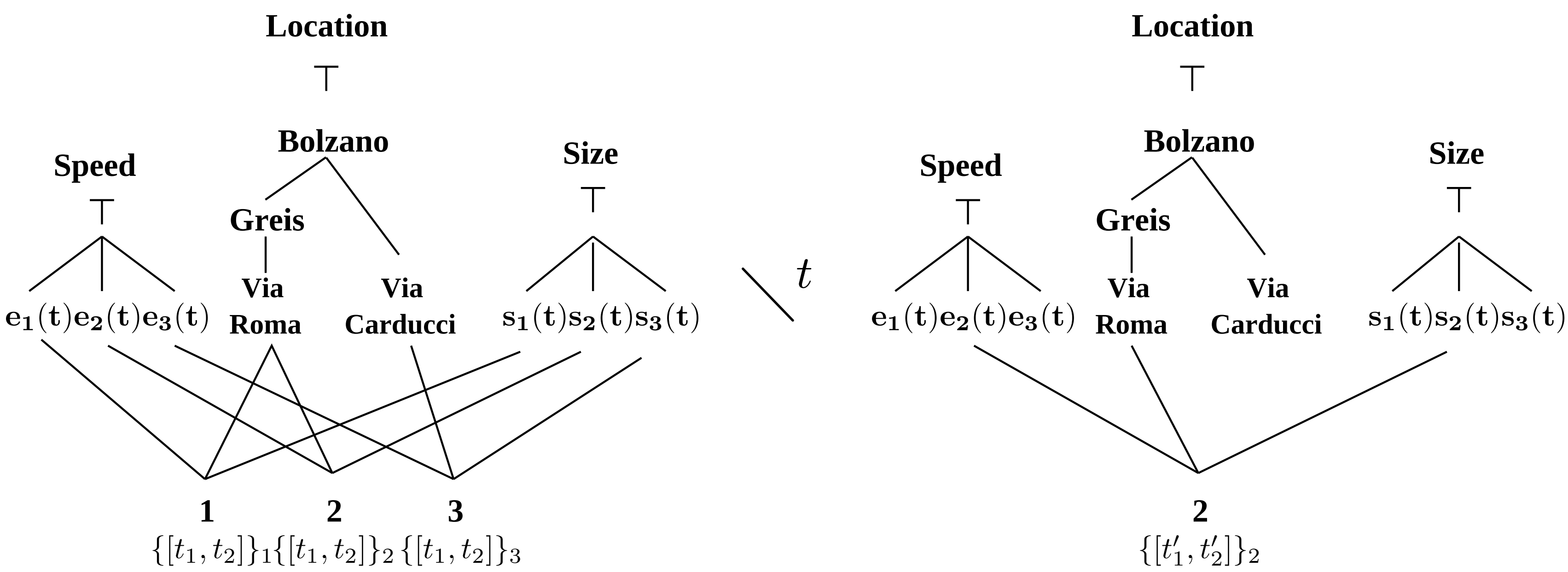
Difference (1/2)

- Difference operator takes difference of facts of two multidimensional objects



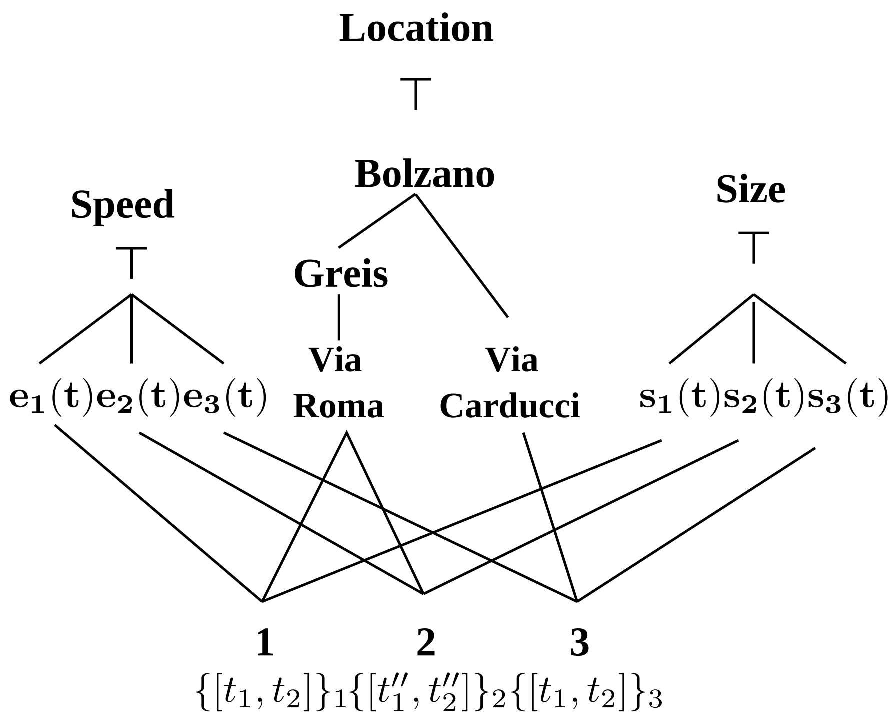
Difference (1/2)

- Difference operator takes difference of facts of two multidimensional objects



Difference (1/2)

- Difference operator takes difference of facts of two multidimensional objects



Difference (2/2)

Provided that $\mathcal{S}_1 = \mathcal{S}_2$, the *CT difference* operator, $\setminus^t$, is defined as: $M_1 \setminus^t M_2 = (\mathcal{S}_1, F', D_1, R')$, where

$$\begin{aligned} F' = \{(f, T_f) | & ((f, T_f) \in F_1 \wedge (f, T'_f) \notin F_2) \vee \\ & ((f, T_f) \in F_1 \wedge (f, T'_f) \in F_2 \wedge \\ & T_f \neq T'_f \wedge T_f = T_f - T'_f)\} \\ R' = \{R'_i&, i = 1, \ldots, n\}, \\ R'_i = \{(f, e(t)) &| (f, T_f) \in F' \wedge (f, e(t)) \in R_{1i}\} \end{aligned}$$

Identity-based Join (1/2)

- Identity-based Join (proposed by Pedersen) joins those facts of two multidimensional objects which satisfy join predicate and their valid time overlap.

Given a predicate $p(f_1, f_2) \in \{f_1 = f_2, f_1 \neq f_2, true\}$, the *CT identity-based join* operator, $\bowtie$, is defined as follows: $M_1 \bowtie [p] M_2 = (\mathcal{S}', F', D', R')$, (mathematical notation omitted)

Identity-based Join (1/2)

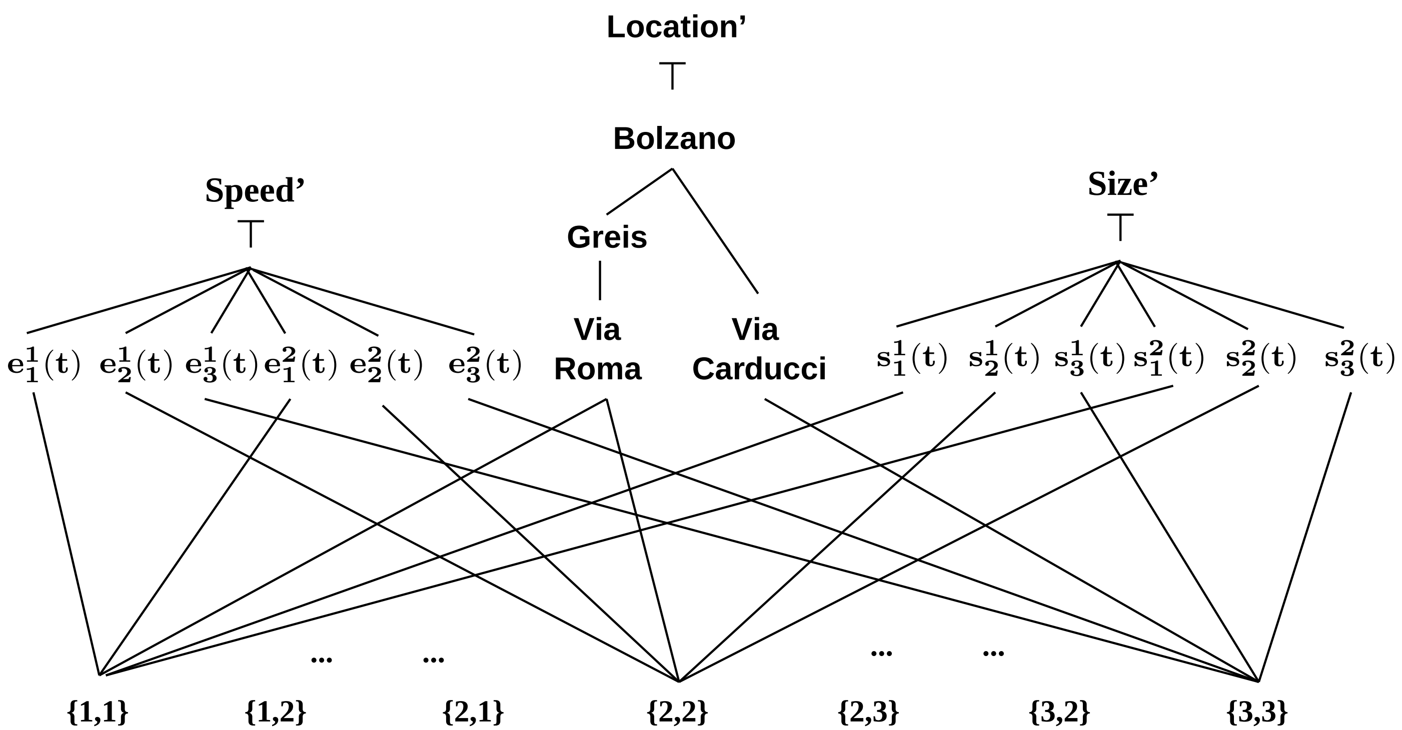
- Identity-based Join (proposed by Pedersen) joins those facts of two multidimensional objects which satisfy join predicate and their valid time overlap.

Given a predicate $p(f_1, f_2) \in \{f_1 = f_2, f_1 \neq f_2, true\}$, the *CT identity-based join* operator, $\bowtie$, is defined as follows: $M_1 \bowtie [p] M_2 = (\mathcal{S}', F', D', R')$, (mathematical notation omitted)

- Properties of Identity-based join:
  - When predicate is *true*, we get Cartesian product
  - Value-based join can be expressed by combining Cartesian product, selection and projection
  - Natural join is special case of value based join

Identity-based Join (2/2)

- When predicete is *true* we get Cartesian product



Aggregate Formation (1/3)

- We support sequenced and non-sequenced aggregation
- Aggregation is performed in three steps:
  1. We group by dimension values(GROUP BY clause)
  2. We group facts into given time intervals ($G^t[T]$)
  3. Apply aggregate function for each group of facts

Aggregate Formation (2/3) - Grouping

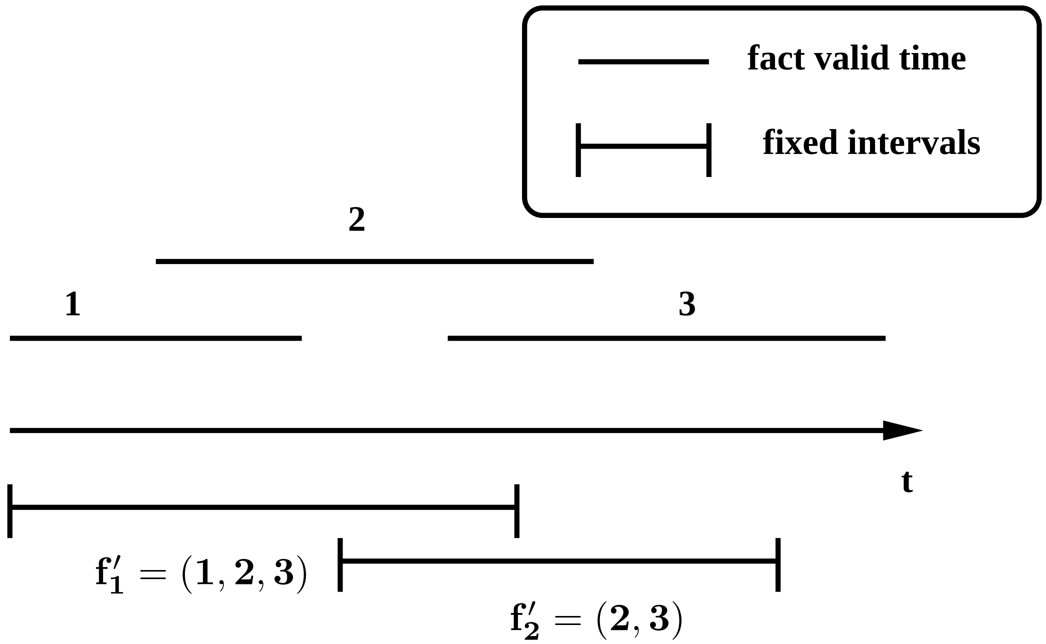
- We perform temporal grouping by *fixed intervals*

Given a time interval, T, and a set of facts, $\overline{F} \subseteq F$, the *CT group* operator, G^t, is defined as:
$G^t[T](\overline{F}) = \{f | (f, T_f) \in \overline{F} \wedge T_f \cap T \neq \emptyset\}$.

Aggregate Formation (2/3) - Grouping

- We perform temporal grouping by *fixed intervals*

Given a time interval, T, and a set of facts, $\overline{F} \subseteq F$, the *CT group* operator, G^t, is defined as:
$G^t[T](\overline{F}) = \{f | (f, T_f) \in \overline{F} \wedge T_f \cap T \neq \emptyset\}$.

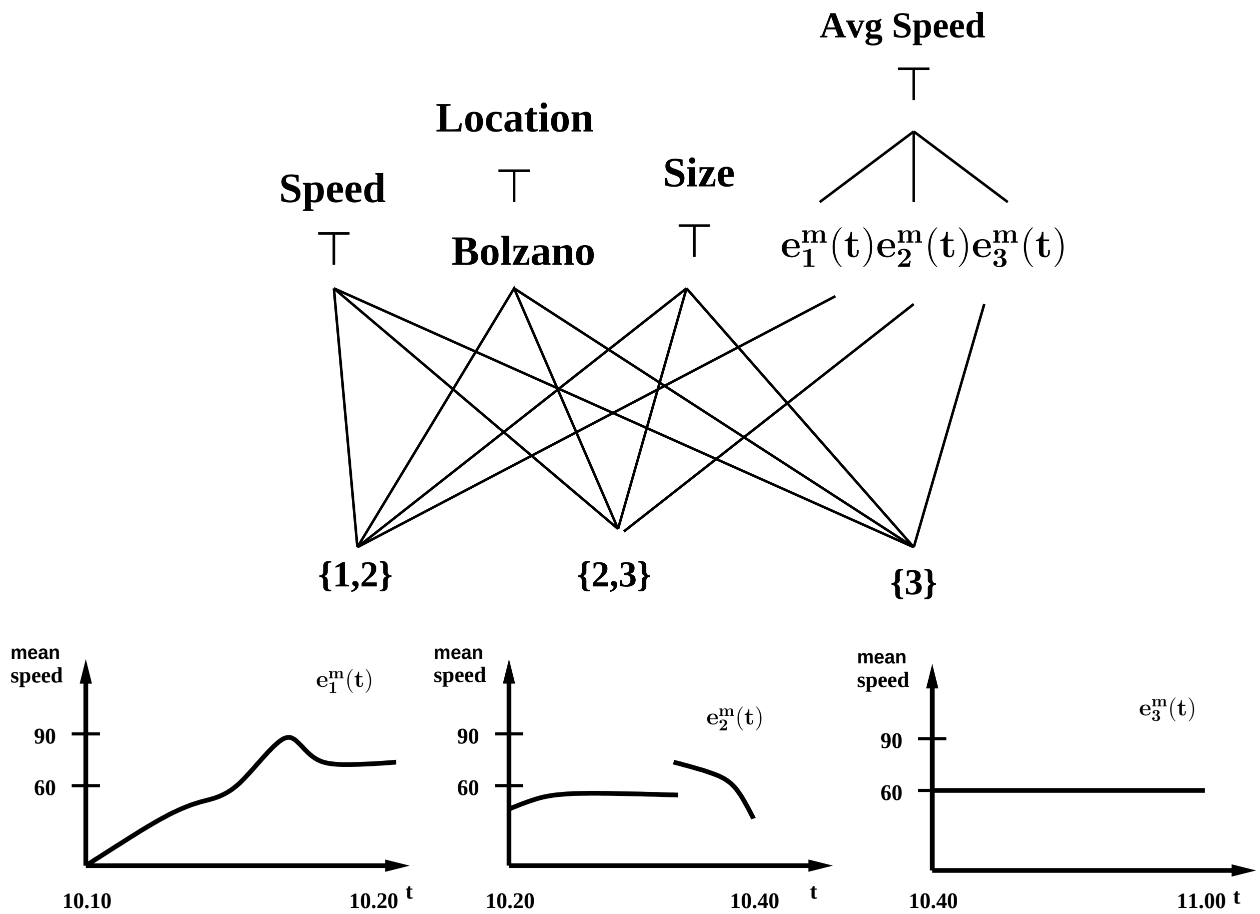


Aggregate formation (3/3) - Example

- $\alpha^t[AvgSpeed, AVG_{speed}, \top, City, \top, T](M)$, where $T = \{[10.00; 10.20], [10.20; 10.40], [10.40, 11.00]\}$.

Aggregate formation (3/3) - Example

- $\alpha^t[AvgSpeed, AVG_{speed}, \top, City, \top, T](M)$, where $T = \{[10.00; 10.20], [10.20; 10.40], [10.40, 11.00]\}$.



Expressive power of CT-OLAP algebra

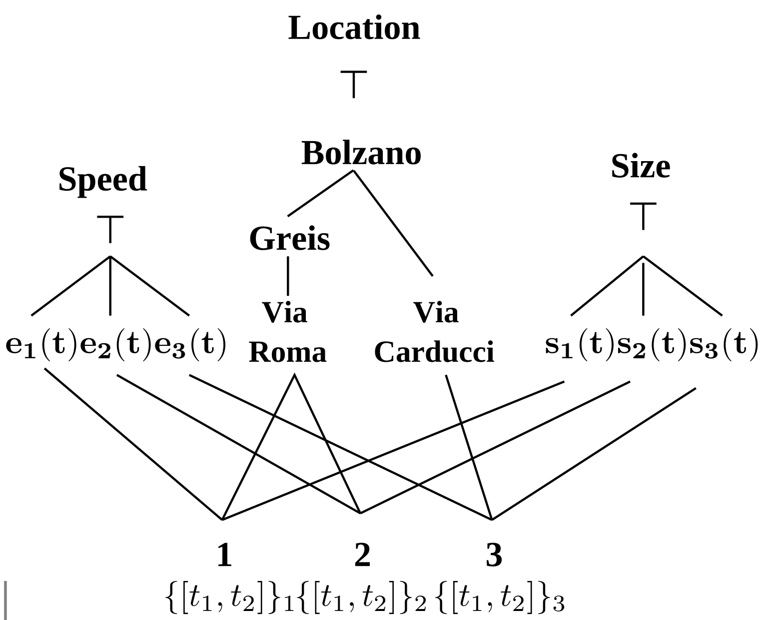
- Goal was to make algebra at least as powerful as BCDM algrbea
  - We support two different facts with same dimension values
  - BCDM does not support two different tuples with same attribute values - does coalescing

Expressive power of CT-OLAP algebra

- Goal was to make algebra at least as powerful as BCDM algrbea
  - We support two different facts with same dimension values
  - BCDM does not support two different tuples with same attribute values - does coalescing
- CT-OLAP model and algebra with coalescing is at least as powerful as bitemporal conceptual data model (BCDM) and its algebra, without transaction time support, extended with aggregate formation operator
- Intuitively, continuity makes CT-OLAP algebra more powerful than BCDM, since continuous functions can not be directly represented in BCDM model

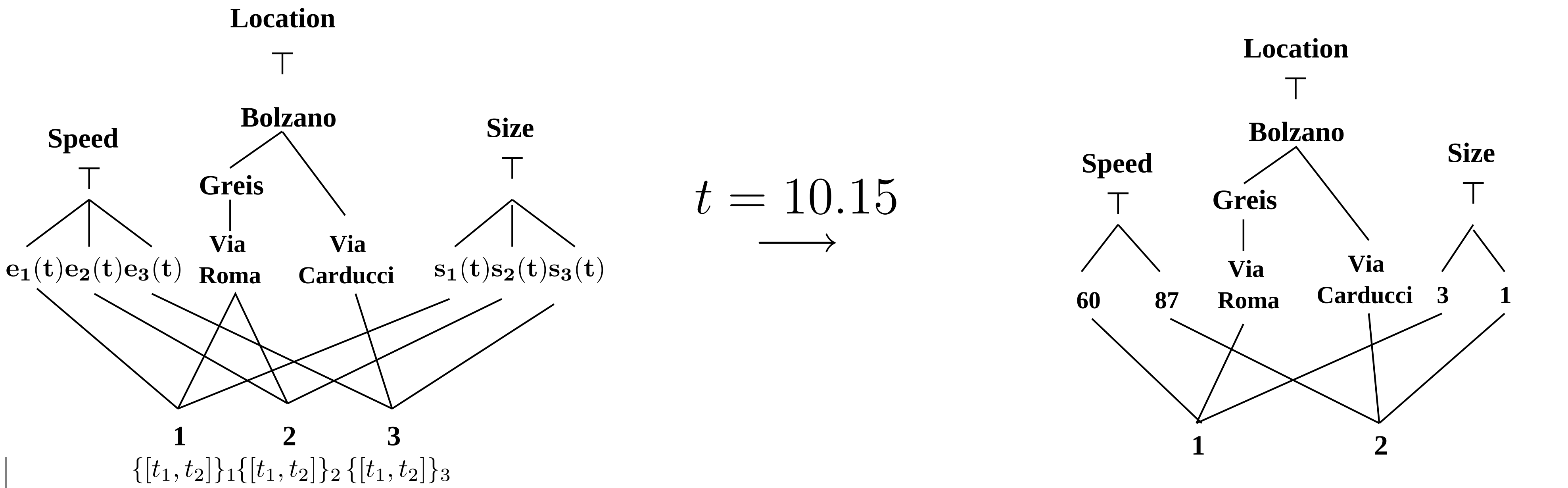
Relation with Pedersen's Model

- Snapshot operator returns a snapshot of a given object at a given instance of time. The resulting object is a multidimensional object defined in terms of Pedersen's model.



Relation with Pedersen's Model

- Snapshot operator returns a snapshot of a given object at a given instance of time. The resulting object is a multidimensional object defined in terms of Pedersen's model.
- We apply the snapshot operator on M with $t = 10.15$



Outline

- Problem
- Current Technology
- Our approach
  - Model Extension
  - Algebra Extension
- Implementation
- Conclusions and Future Work

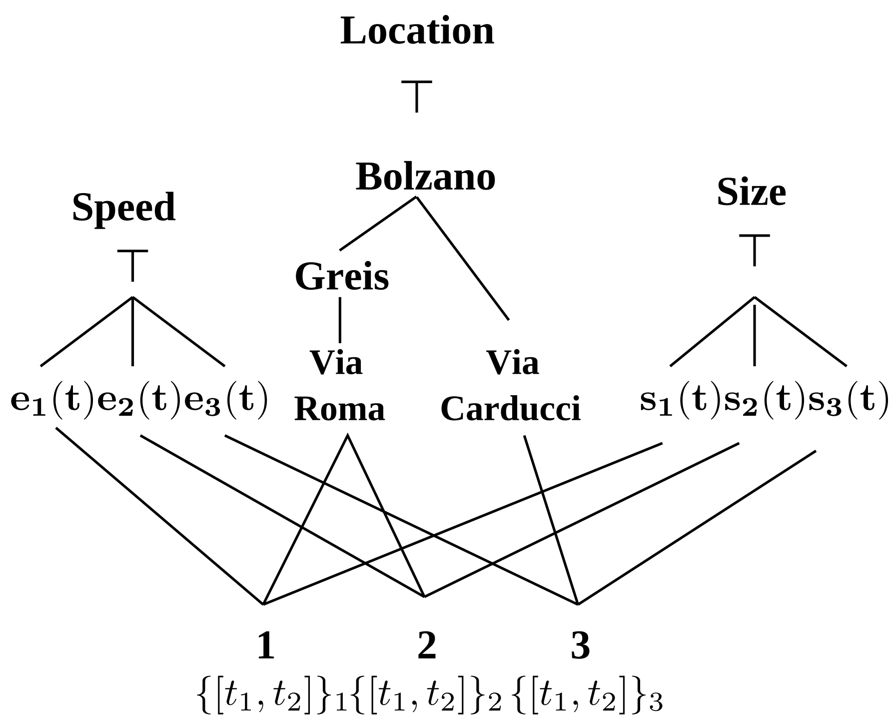
Implementation (1/2)

- We are implementing our model and algebra in Secondo Database management system [5]
- Secondo is a database management system that supports continuous changes in continuous time
- We use:
  - Secondo data types - for representing dimension functions
  - Secondo operations - for algebraic operators

Implementation (2/2)

- Fact table in a relational representation of and the extended model

	Speed	Location	Size
1	$e_1(t)$	**Via Roma**	$s_1(t)$
2	$e_2(t)$	**Via Roma**	$s_2(t)$
3	$e_3(t)$	**Via Carducci**	$s_3(t)$



Outline

- Problem
- Current Technology
- Our approach
  - Model Extension
  - Algebra Extension
- Implementation
- Conclusions and Future Work

Summary and Conclusions

- We present a continuous time multidimensional model for temporal OLAP by extending Pedersen's et al. model
- We extend existing algebra to handle continuous dimension values
- Discrete data can be modeled together with continuous data
- Expressive power of CT-OLAP algebra is at least as powerful as algebras of BCDM model

Future Work

- Research perspective
  - Investigation about hierarchies in dimensions with continuous functions
  - To investigate possibility of extending Probabilistic OLAP models to handle continuous time

Future Work

- Research perspective
  - Investigation about hierarchies in dimensions with continuous functions
  - To investigate possibility of extending Probabilistic OLAP models to handle continuous time
- Implementation perspective
  - Fully implement relational representation of extended model and algebra
  - Extensive experimental evaluation on a real world data

The End

1 J. Eder and C. Koncilia. Changes of Dimension Data in Temporal Warehouses. In *DaWaK*, pp. 284-293, 2001.

2 P. Chamoni and S. Stock. Temporal Structures in Data Warehousing. In *DaWaK*, pp. 353–358, 1999.

3 A. O. Mendelzon and A. A. Vaisman. Temporal Queries in Olap. In *VLDB*, pp. 242–253, 2000.

4 C. S. Jensen and M. D. Soo and R. T. Snodgrass. Unifying Temporal Data Models via a Conceptual Model". "Information Systems", pp.513-547, 1994

5 R.H. Güting et al., Secondo: An Extensible DBMS Platform for Research Prototyping and Teaching. In *ICDE*, pp.1115–1116, 2005.

Appendix (1/2)

- Mathematical notation of identity-based join operator

Given a predicate $p(f_1, f_2) \in \{f_1 = f_2, f_1 \neq f_2, true\}$, the *CT identity-based join* operator, $\bowtie$, is defined as follows: $M_1 \bowtie [p] M_2 = (\mathcal{S}', F', D', R')$, where $\mathcal{S}' = (\mathcal{F}', \mathcal{D}')$, $\mathcal{F}' = \mathcal{F}_1' \times \mathcal{F}_1'$, $\mathcal{D}' = \mathcal{D}_1 \cup \mathcal{D}_2$, $F' = \{(f_1, f_2, t_f')|(f_1, t_f^1) \in F_1 \wedge (f_2, t_f^2) \in F_2 \wedge p(f_1, f_2) \wedge t_f^1 \cap t_f^2 \neq \emptyset \wedge t_f' = t_f^1 \cap t_f^2\}$, $D' = D_1 \cup D_2$, $R' = \{R_i', i = 1, \ldots, n_1 + n_2\}$, and $R_i' = \{(f', e)|f' = (f_1, f_2) \wedge f' \in F' \wedge ((i \leq n_1 \wedge (f_1, e) \in R_{1i}) \vee (i > n_1 \wedge (f_2, e) \in R_{2(i-n_1)}))\}$.

Appendix (2/2)

- Mathematical notation of aggregate formation operator

Aggregate formation operator is defined as follows
$\alpha[D_{n+1}, g, C_1, \ldots, C_n](MO) = (\mathcal{S}', F', D', R')$, where:
$\mathcal{S}' = (\mathcal{F}', \mathcal{D}')$, $\mathcal{F}' = 2^{\mathcal{F}}$, $\mathcal{D}' = \{\mathcal{T}_i', i = 1, ..., n\} \cup \{\mathcal{T}_{n+1}\}$,
$\mathcal{T}_i' = (\mathcal{C}_i', \sqsubset_{\mathcal{T}_i'}, \top_{\mathcal{T}_i'}, \bot_{\mathcal{T}_i'})$, $\mathcal{C}_i' = \{\mathcal{C}_{ij} \in \mathcal{T}_i | \mathcal{C}_i = \mathcal{C}_{ij} \vee \mathcal{C}_i \sqsubset_{\mathcal{T}_i} \mathcal{C}_{ij}\}$,
$\sqsubset'_{\mathcal{T}_i} = \sqsubset_{\mathcal{T}_i|\mathcal{C}_i'}$, $\top_{\mathcal{T}_i'} = \top_{\mathcal{T}_i}$, $\bot_{\mathcal{T}_i'} = \mathcal{C}_i$,
$F' = \{Group(\vec{e}) | \vec{e} \in C_1 \times \cdots \times C_n \wedge Group(\vec{e}) \neq \emptyset\}$,
$D' = \{D_i', i = 1, ..., n\} \cup \{D_{n+1}\}$, $D_i' = (C_i', \sqsubset_i')$,
$C_i' = \{C_{ij}' \in D_i | \mathcal{C}_{ij}') \in \mathcal{C}_i'\}$, $\sqsubset_i' = \sqsubset_{i|D_i'}$,
$R' = \{R_i', i = 1, ..., n\} \cup \{R_{n+1}'\}$,
$R_i' = \{(f', e_i) | \exists \vec{e} \in C_1 \times \cdots \times C_n (f' = Group(\vec{e}) \wedge f' \in F')\}$,
and $R_{n+1}' = \{(f', g(f')) | f' \in F'\}$