Warehousing and Mining Massive RFID Data Sets

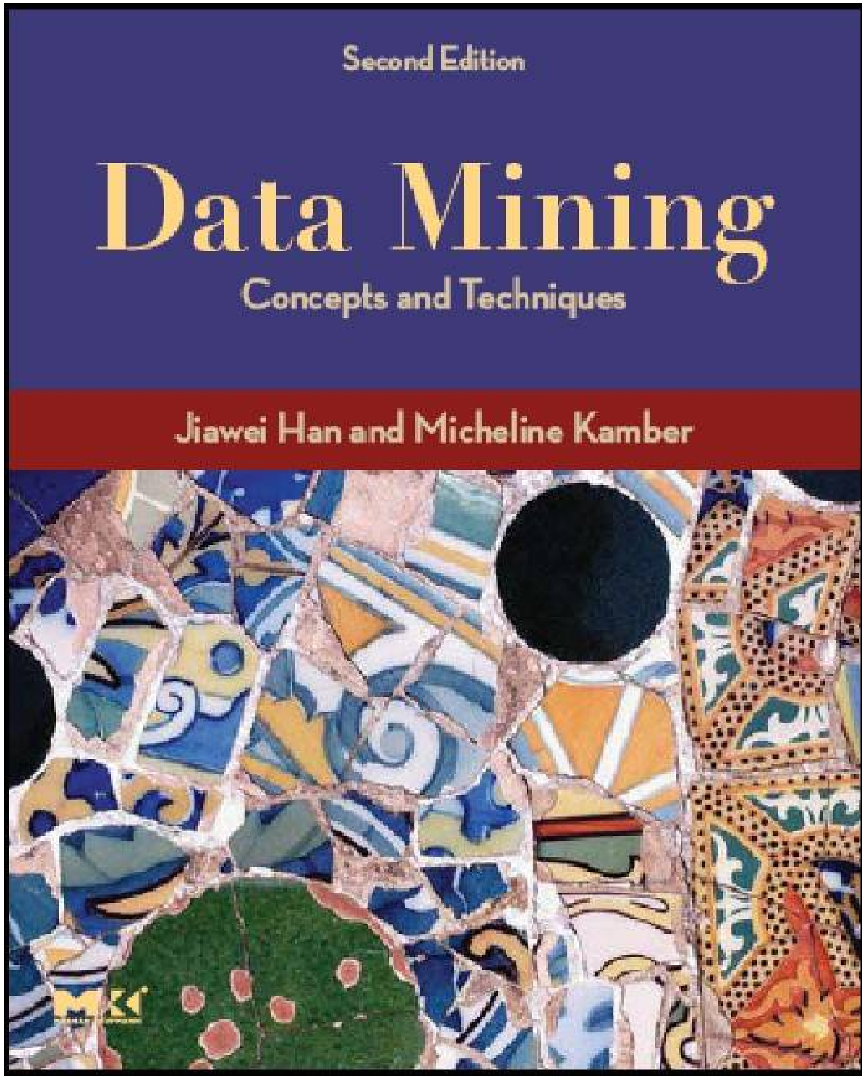
Jiawei Han
Department of Computer Science
University of Illinois at Urbana-Champaign
www.cs.uiuc.edu/~hanj

Our Recent Work on Data Mining & Applications

- Pattern mining, pattern usage, and pattern understanding
- Information network analysis
- Stream data mining
- Mining moving object, spatiotemporal, and multimedia data
- Biological data mining
- Text and Web mining
- Data mining for software engineering and computer systems
- Cube-oriented ranking and multidimensional analysis
- Warehousing and mining RFID data

Data Mining: Concepts and Techniques, 2ed. 2006

- Mining stream, time-series, and sequence data
  - Mining data streams
  - Mining time-series data
  - Mining sequence patterns in transactional databases
  - Mining sequence patterns in biological data
- Graph mining, social network analysis, and multi-relational data mining
  - Graph mining
  - Social network analysis
  - Multi-relational data mining
- Mining Object, Spatial, Multimedia, Text and Web data
  - Mining object data
  - Spatial and spatiotemporal data mining
  - Multimedia data mining
  - Text mining
  - Web mining



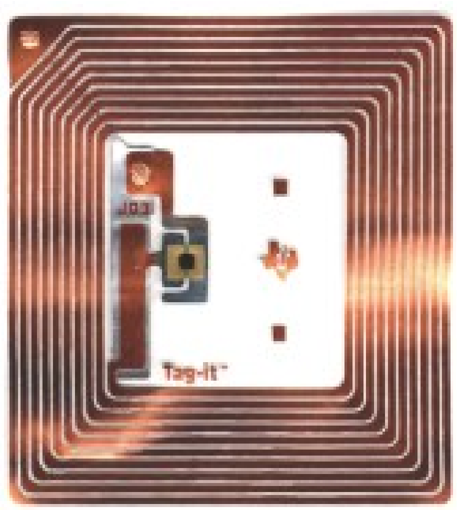
Outline

- Introduction to RFID Technology
- Why RFID Data Warehousing and Mining?
- RFID Data Warehousing
- Mining RFID Data Sets
- Conclusions

What Is RFID?

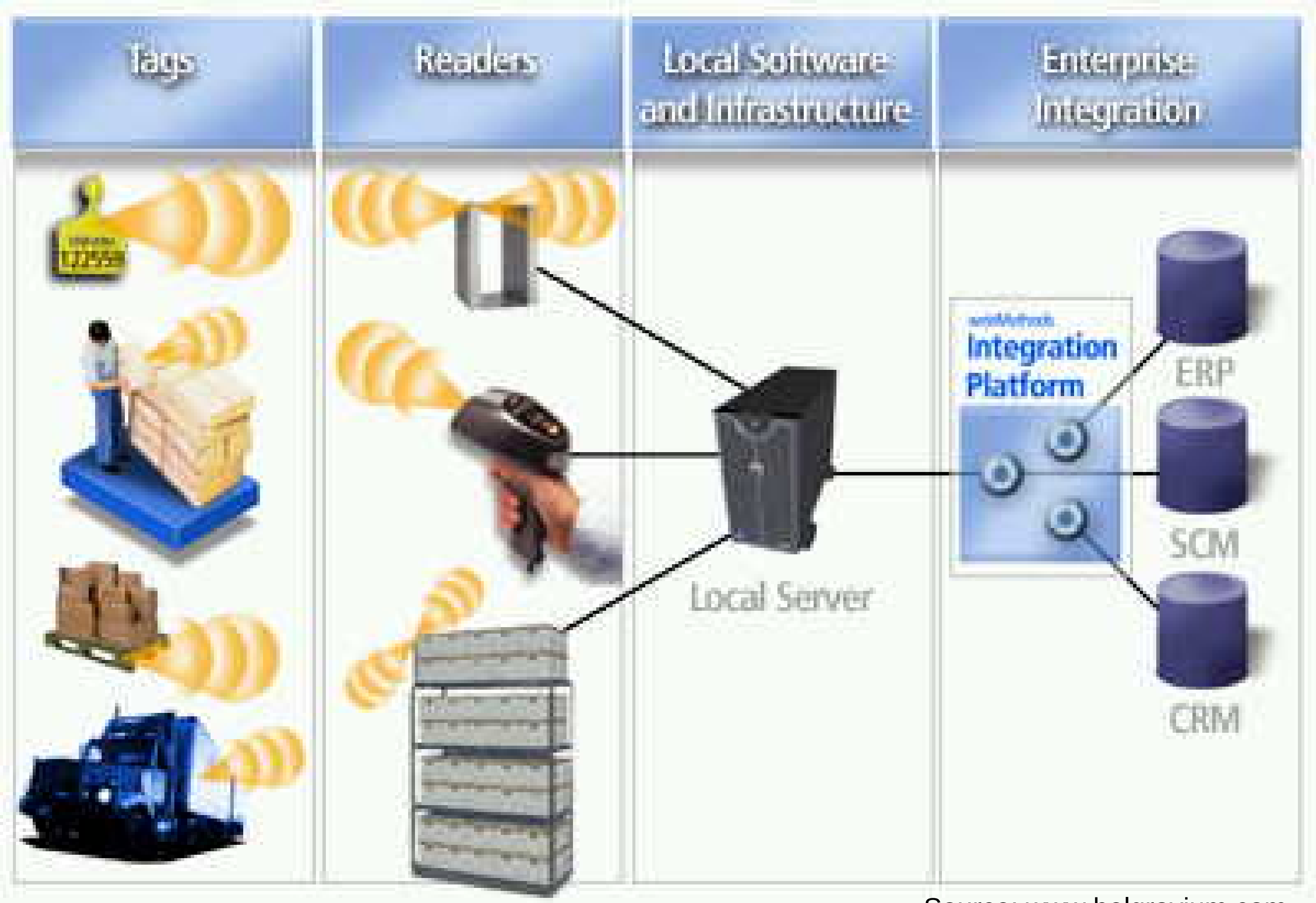
- Radio Frequency Identification (RFID)
  - Technology that allows a sensor (reader) to read, from a distance, and without line of sight, a unique electronic product code (EPC) associated with a tag

Tag



Reader

RFID System (Tag, Reader, Database)



Source: www.belgravium.com

Broad Applications of RFID Technology

- **Supply Chain Management**: real-time inventory tracking
- **Retail**: Active shelves monitor product availability
- **Access control**: toll collection, credit cards, building access
- **Airline luggage management**: reduce lost/misplaced luggages
- **Medical**: Implant patients with a tag that contains their medical history
- **Pet identification**: Implant RFID tag with pet owner information

Outline

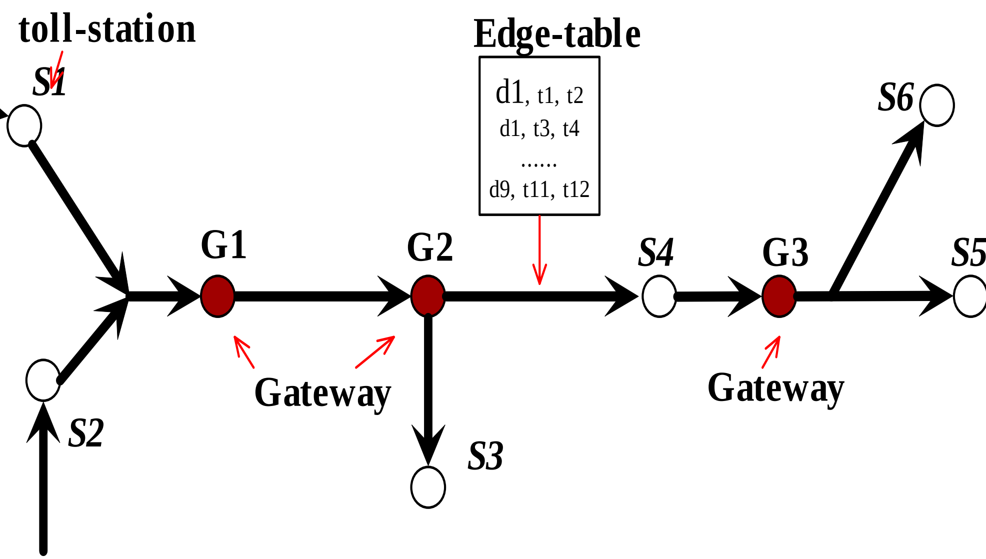
- Introduction to RFID Technology
- Why RFID Data Warehousing and Mining?
- RFID Data Warehousing
- Mining RFID Data Sets
- Conclusions

Challenges of RFID Data Sets

- Data generated by RFID systems is enormous (peta-bytes in scale!) due to redundancy and low level of abstraction
  - Walmart is expected to generate 7 terabytes of RFID data per day
- Data analysis requirements
  - Highly compact summary of the data
  - OLAP operations on multi-dimensional view of the data
  - Preserving the path structures of RFID data for analysis
  - Efficiently drilling down to individual tags when an interesting pattern is discovered

RFID Data Warehouse Modeling

- Three models in typical RFID applications
  - Bulky movements: supply-chain management
  - Scattered movements: E-pass tollway system
  - No movements: fixed location sensor networks
- Different applications may require different data warehouse systems
- Our discussion will focus on *bulky movements*



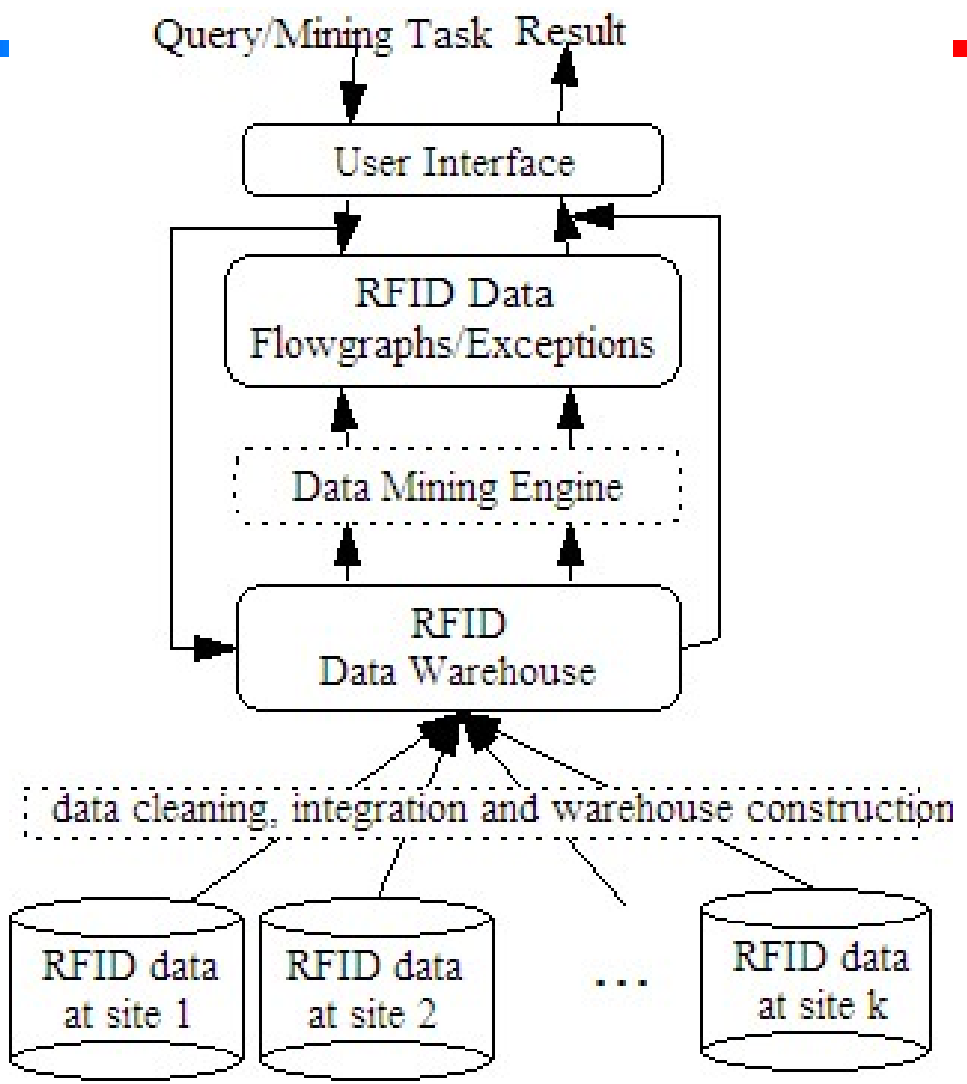
Why RFID-Warehousing?

- Lossless compression for bulky movement data
  - Significantly reduce the size of the RFID data set by redundancy removal and grouping objects that move and stay together
- Data cleaning: reasoning based on more complete info
  - Multi-reading, miss-reading, error-reading, bulky movement, …
- Multi-dimensional summary, multiple views
  - Multiple dimensional view: Product, location, time, …
  - **Store manager**: Check item movements from the backroom to different shelves in his store
  - **Region manager**: Collapse intra-store movements and look at distribution centers, warehouses, and stores

RFID OLAP, Path Query and Mining

- Warehousing supports FRID query processing
  - Support for OLAP: roll-up, drill-down, slice, and dice
  - Path query: New to RFID-Warehouse, about the *structure of paths*
    - What products that go through quality control have shorter paths?
    - What locations are common to the paths of a set of defective auto-parts?
    - Identify containers at a port that have deviated from their historic paths
- FRID data mining
  - Find trends, outliers, frequent, sequential, flow patterns, …

RFID Warehouse Architecture



Example: A Supply Chain Store

- A retailer with 3,000 stores, selling 10,000 items a day per store
- Each item moves 10 times on average before being sold
  - Movement recorded as (EPC, location, second)
- Data volume: 300 million tuples per day (after redundancy removal)
- OLAP query: Costly to answer if scanning 1 billion tuples
  - Avg time for outwear items to move from warehouse to checkout counter in March 2006?
- Mining query:
  - Is there a correlation between the time spent at transportation and the milk in store S rotten?

Outline

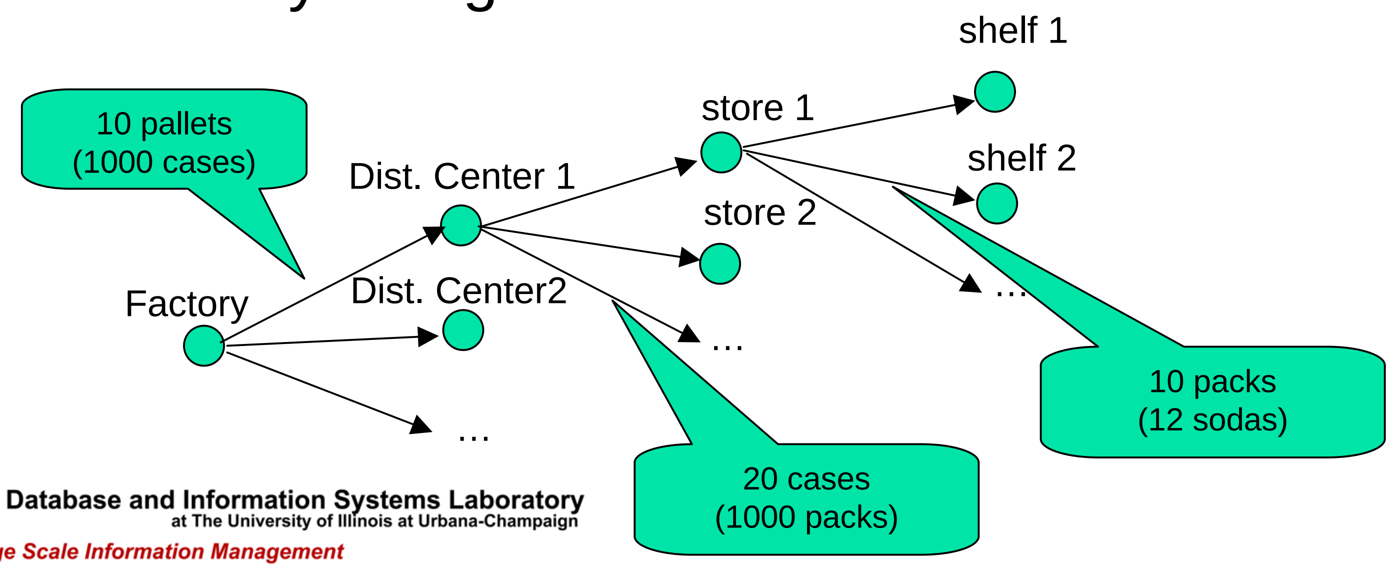
- Introduction to RFID Technology
- Why RFID Data Warehousing and Mining?
- RFID Data Warehousing
- Mining RFID Data Sets
- Conclusions

Cleaning of RFID Data Records

- Raw Data
  - (EPC, location, time)
  - Duplicate records due to multiple readings of a product at the same location
  - (r_1, l_1, t_1) (r_1, l_1, t_2) ... (r_1, l_1, t_{10})
- Cleansed Data: Minimal information to store and removal of raw data
  - (EPC, Location, time_in, time_out)
  - (r_1, l_1, t_1, t_{10})
- Warehousing can help fill-up missing records and correct wrongly-registered information

Data Compression with GID

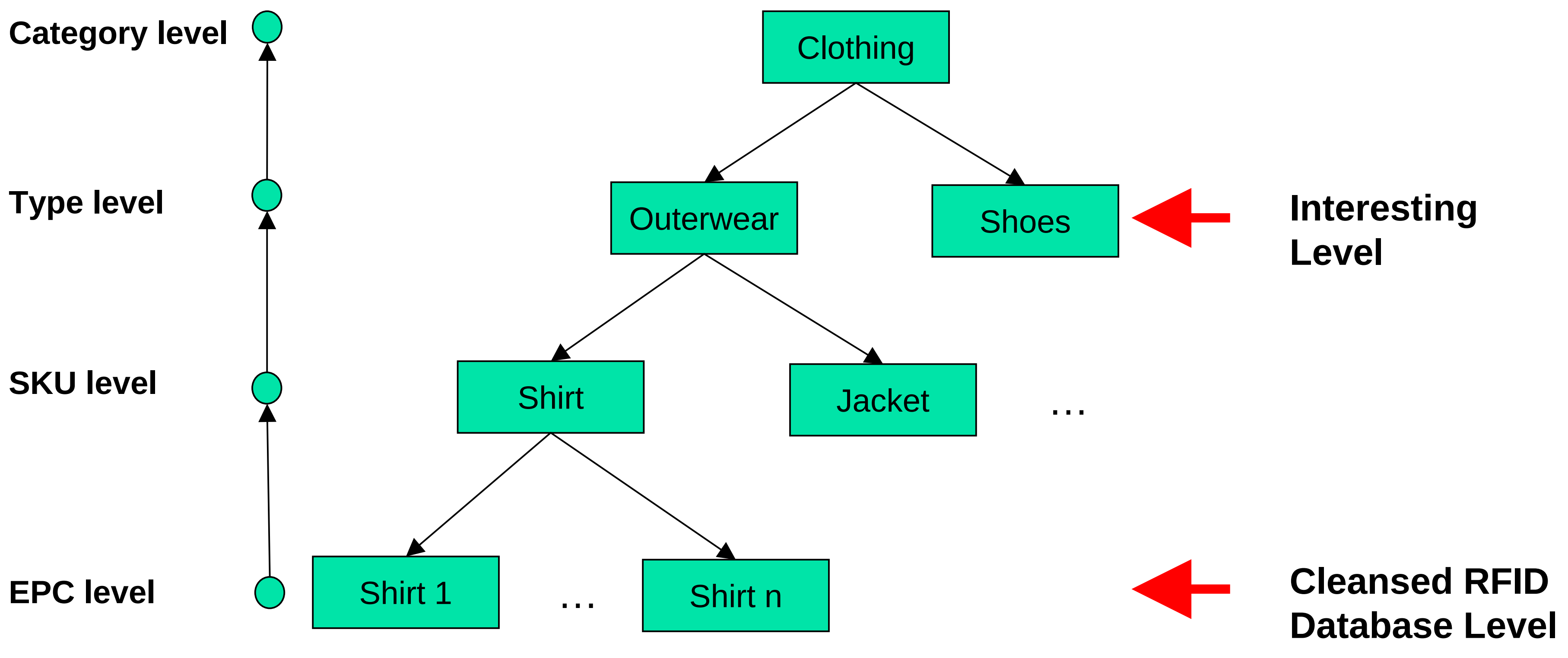
- Bulky object movements
  - Objects often move and stay together
  - If 1000 packs of soda stay together at the distribution center, register a single record
    - (GID, distribution center, time_in, time_out)
  - GID is a generalized identifier that represents the 1000 packs that stayed together at the distribution center



Compression by Data/Path Generalization

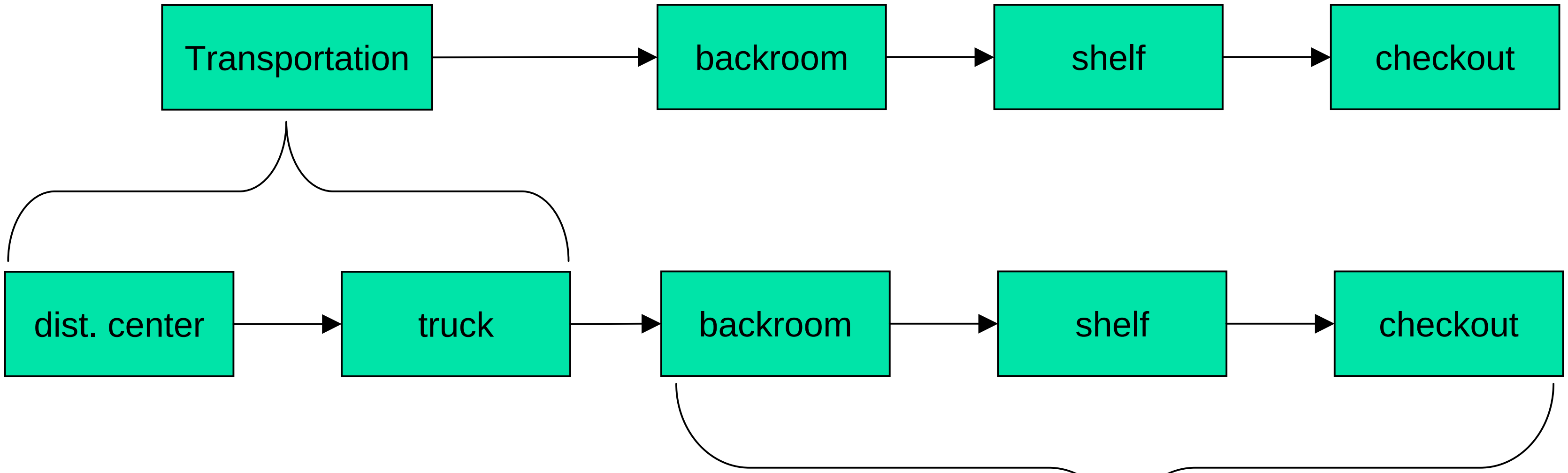
- **Data generalization**
  - Analysis usually takes place at a much higher level of abstraction than the one present in raw RFID data
  - Aggregate object movements into fewer records
    - If interested in time at the day level, merge records at the minute level into records at the hour level
- **Path generalization**: Merge and/or collapse path segments
  - Uninteresting path segments can be ignored or merged
  - Multiple item movements within the same store may be uninteresting to a regional manager and thus merged

Path-Independent Data Generalization

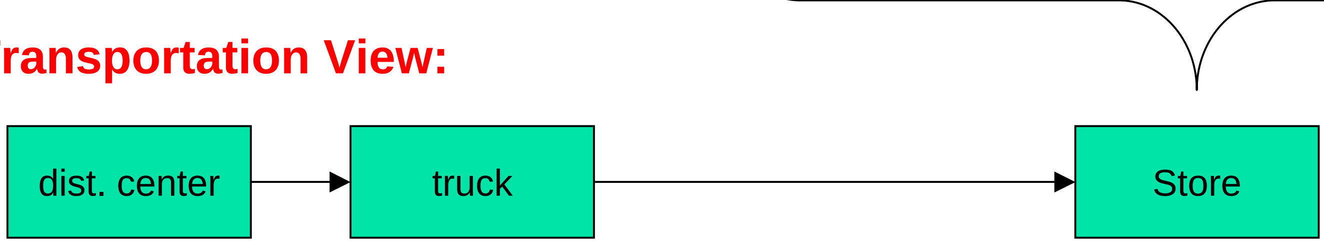


Path Generalization

Store View:



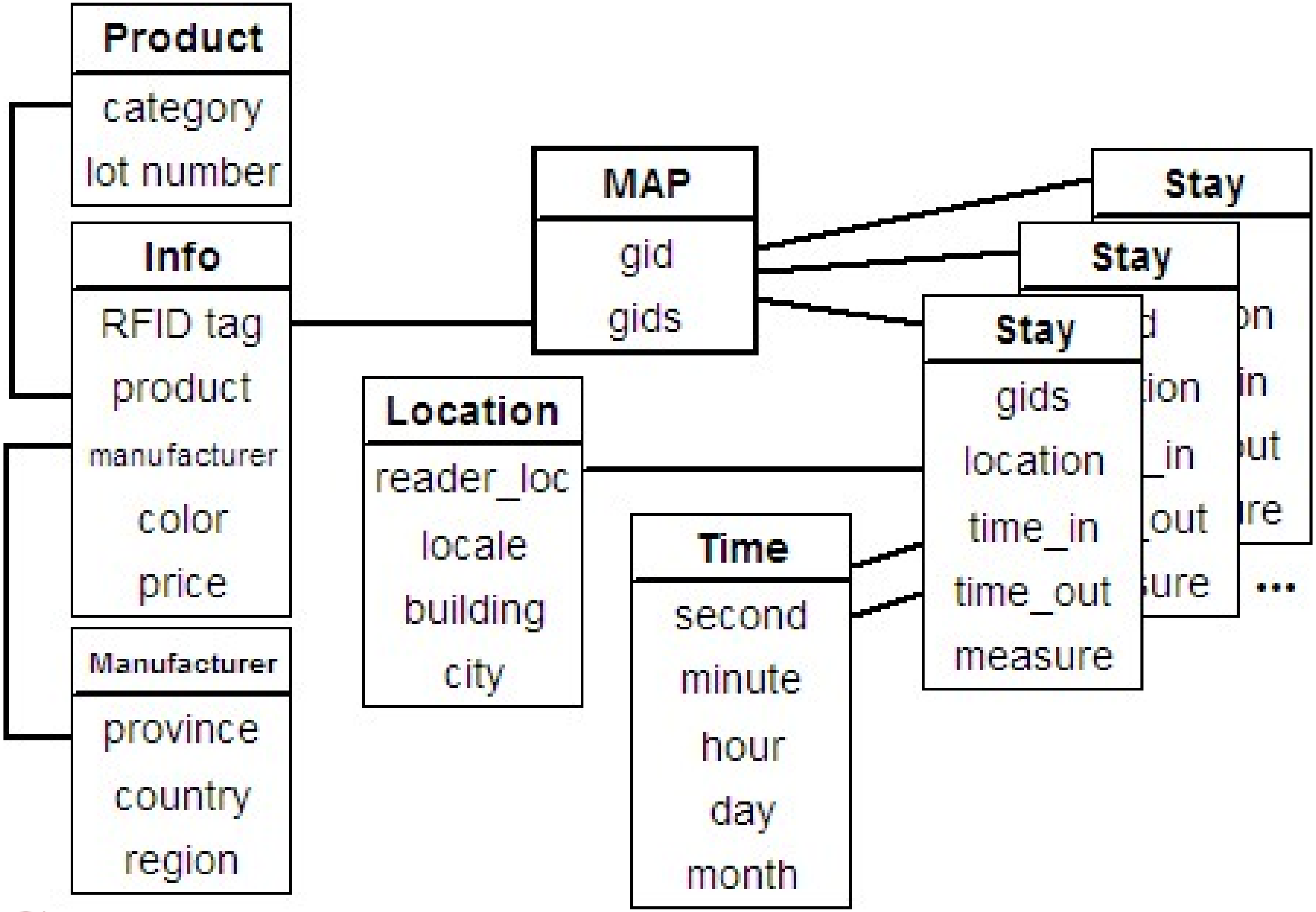
Transportation View:



Why Not Using Traditional Data Cube?

- Fact Table: (EPC, location, time_in, time_out)
- Aggregate: A measure at a single location
  - e.g., what is the average time that milk stays in the refrigerator in Illinois stores?
- What is missing?
  - Measures computed on items that **travel through** a series of locations
  - e.g., what is the average time that milk stays at the refrigerator in Champaign when coming from farm A, and Warehouse B?
- Traditional cubes **miss the path structure of the data**

RFID-Cube Architecture



Three RFID-Cuboids

- **Stay Table:** (GIDs, location, time_in, time_out: measures)
  - Records information on items that stay together at a given location
  - If using record transitions: difficult to answer queries, lots of intersections needed
- **Map Table:** (GID, <GID_1,..,GID_n>)
  - Links together stages that belong to the same path. Provides additional: compression and query processing efficiency
  - High level GID points to lower level GIDs
  - If saving complete EPC Lists: high costs of IO to retrieve long lists, costly query processing
- **Information Table:** (EPC list, attribute 1,...,attribute n)
  - Records path-independent attributes of the items, e.g., color, manufacturer, price

RFID-Cuboid Example

Cleansed RFID Database

epc	loc	t_in	t_out
r1	l1	t1	t10
r1	l2	t20	t30
r2	l1	t1	t10
r2	l3	t20	t30
r3	l1	t1	10
r3	l4	t15	t20

Stay Table

gids	loc	t_in	t_out
g1	l1	t1	t10
g1.1	l2	t20	t30
g1.2	l4	t15	t20

Map Table

gid	gids
g1	g1.1,g1.2
g1.1	r1,r2
g1.2	r3

Benefits of the Stay Table (I)

Query: What is the average time that items stay at location l ?

Transition Grouping

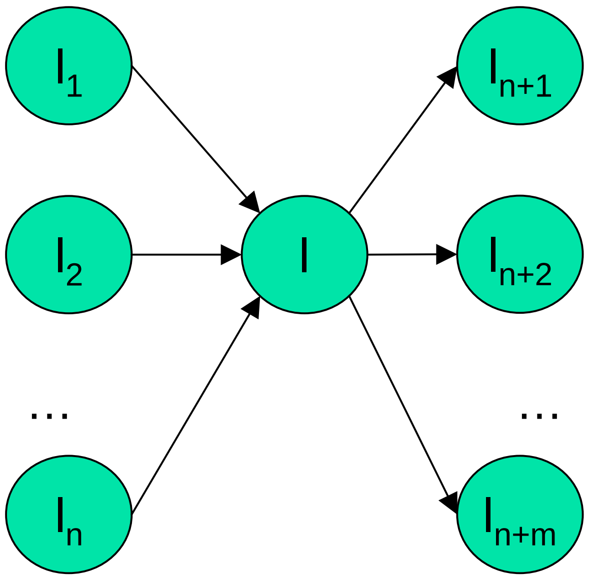
- Retrieve all transitions with destination = l
- Retrieve all transitions with origin = l
- Intersect results and compute average time
- IO Cost: n + m retrievals

Prefix Tree

- Retrieve n records

Stay Grouping

- Retrieve stay record with location = l
- IO Cost: 1



Benefits of the Stay Table (II)

Query: How many boxes of milk traveled through the locations l1, l7, l13?

With Cleansed Database

Strategy:

- Retrieve itemsets for locations l1, l7, l13
- Intersect itemsets

IO Cost:

- One IO per item in locations l1 or l7 or l13

Observation:

- Very costly, we retrieve records at the individual item level

(r1, l1, t1, t2)
(r1, l2, t3, t4)
…
(r2, l1, t1, t2)
(r2, l2, t3, t4)
…
(rk, l1, t1, t2)
(rk, l2, t3, t4)

With Stay Table

Strategy:

- Retrieve the gids for l1, l7, l13
- Intersect the gids

IO Cost:

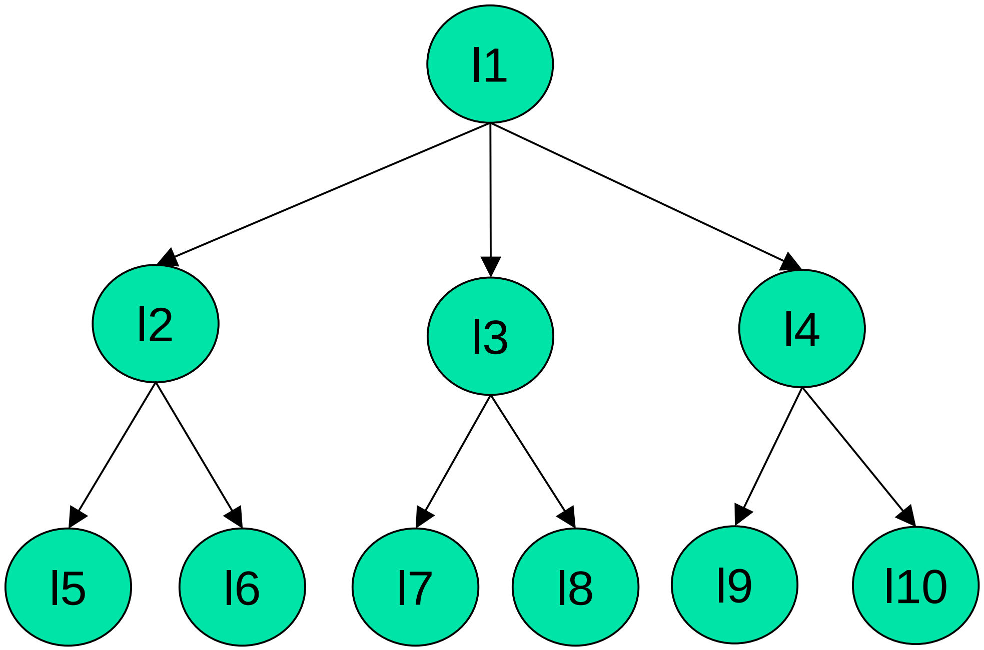
- One IO per GID in locations l1, l7, and l13

Observation:

- Retrieve records at the group level and thus greatly reduce IO costs

(g1, l1, t1, t2)
(g2, l2, t3, t4)
…

Benefits of the Map Table

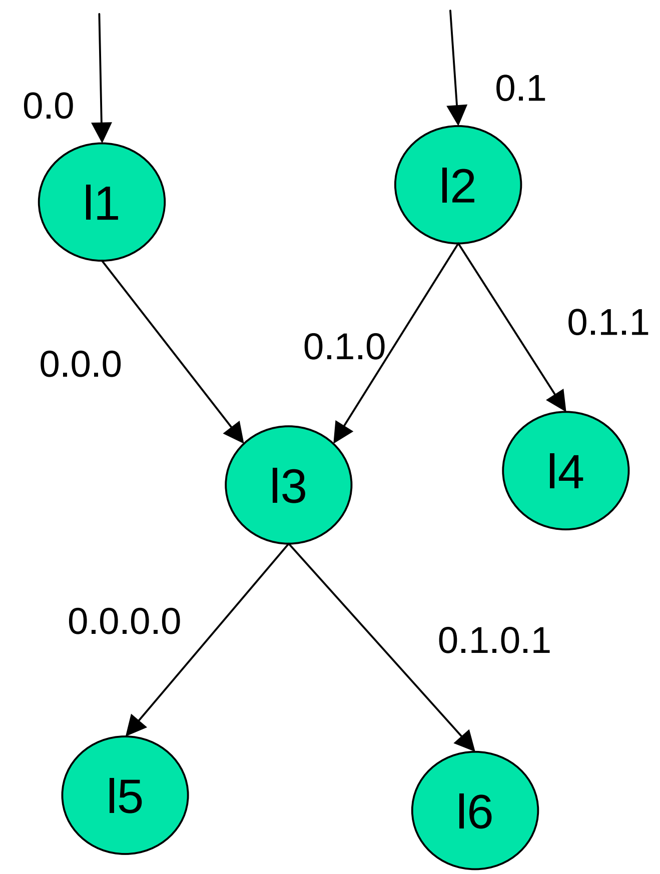


$\{r_1,\ldots,r_i\}$ $\{r_{i+1},\ldots,r_j\}$ $\{r_{j+1},\ldots,r_k\}$ $\{r_{k+1},\ldots,r_l\}$ $\{r_{l+1},\ldots,r_m\}$ $\{r_{m+1},\ldots,r_n\}$

#EPCs	#GIDs
n	1
n	3
n	6
3n	10+n

Path-Dependent Naming of GIDs

- Assign to each GID a unique identifier that encodes the path traversed by the items that it points to
- Path-dependent name: Makes it easy to detect if locations form a path



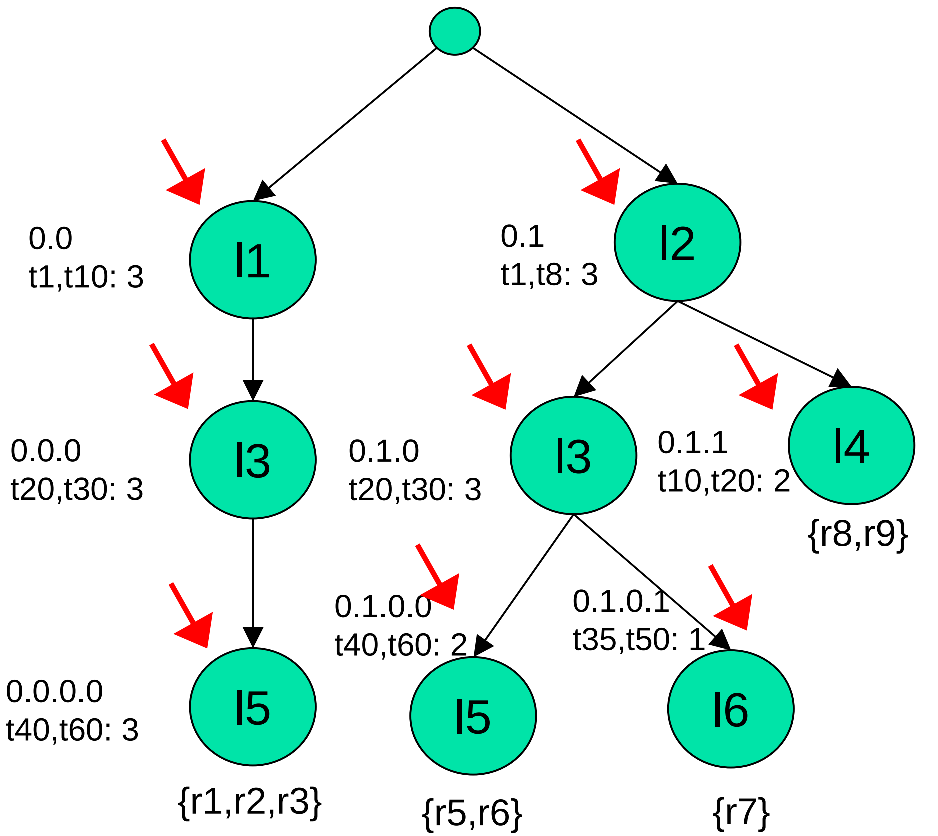
RFID-Cuboid Construction Algorithm

1. Build a prefix tree for the paths in the cleansed database
2. For each node, record a separate measure for each group of items that share the same leaf and information record
3. Assign GIDs to each node:

   GID = parent GID + unique id
4. Each node generates a stay record for each distinct measure
5. If multiple nodes share the same location, time, and measure, generate a single record with multiple GIDs

RFID-Cube Construction

Path Tree



Stay Table

GIDs	loc	t_in	t_out	count
0.0	l1	t1	t10	3
0.0.0 0.1.0	l3	t20	t30	6
0.0.0.0 0.1.0.0	l5	t40	t60	5
0.1	l2	t1	t8	3
0.1.0.1	l6	t35	t50	1
0.1.1	l4	t10	t20	2

RFID-Cube Properties

- The RFID-cuboids can be constructed on a single scan of the cleansed RFID database
- The RFID-cuboid provides lossless compression at its level of abstraction
- The size of the RFID-cuboid is much smaller than the cleansed data
  - In our experiments we get 80% lossless compression at the level of abstraction of the raw data

Query Processing

- Traditional OLAP operations
  - Roll up, drill down, slice, and dice
  - Can be implemented efficiently with traditional optimization techniques, e.g., what is the average time spent by milk at the shelf

$$\sigma_{\text{stay.location = 'shelf', info.product = 'milk'}} (\text{stay} \bowtie_{\text{gid}} \text{info})$$

- Path selection (New operation)
  - Compute an aggregate measure on the tags that travel through a set of locations and that match a selection criteria on path independent dimensions

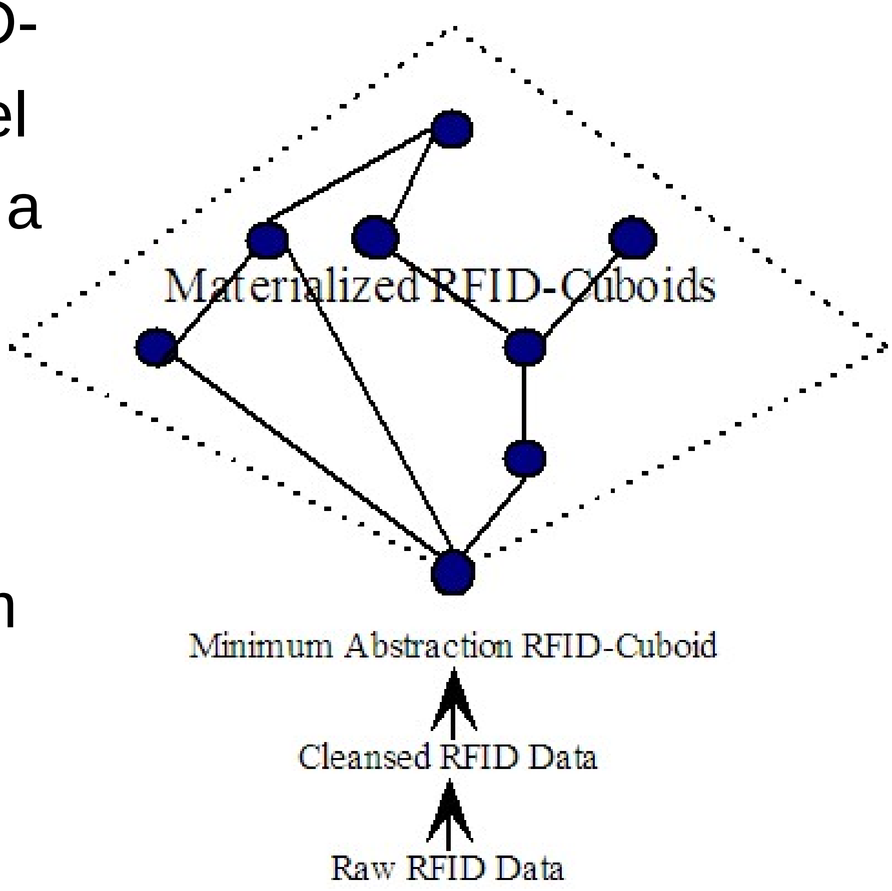
$$q \; \tilde{A} < \sigma_c \; \text{info}, (\sigma_{c_1} \; \text{stage}_1, \ldots, \sigma_{c_k} \; \text{stage}_k) >$$

Query Processing (II)

- Query: What is the average time spent from l3 to l5?
  - GIDs for l3 <0.0.0>, <0.1.0>
  - GIDs for l5 <0.0.0.0>, <0.1.0.1>
  - Prefix pairs: p1: (<0.0.0>,<0.0.0.0>)
    p2: (<0.1.0>,<0.1.0.1>)
- Retrieve stay records for each pair (including intermediate steps) and compute measure
- Savings: No EPC list intersection, remember that each EPC list may contain millions of different tags, and retrieving them is a significant IO cost

From RFID-Cuboids to RFID-Warehouse

- Materialize the lowest RFID-cuboid at the minimum level of abstraction interested to a user
- Materialize frequently requested RFID-cuboids
- Materialization is done from the smallest materialized RFID-Cuboid that is at a lower level of abstraction

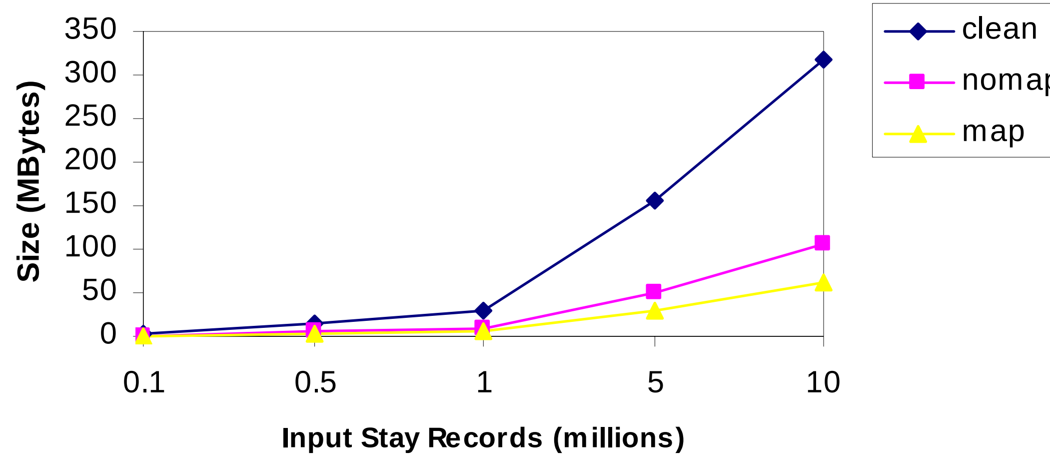


Performance Study: RFID-Cube Compression

Compression vs. Cleansed data size

P = 1000, B = (500,150,40,8,1), k = 5

Lossless compression, cuboid is at the same level of abstraction as cleansed RFID database

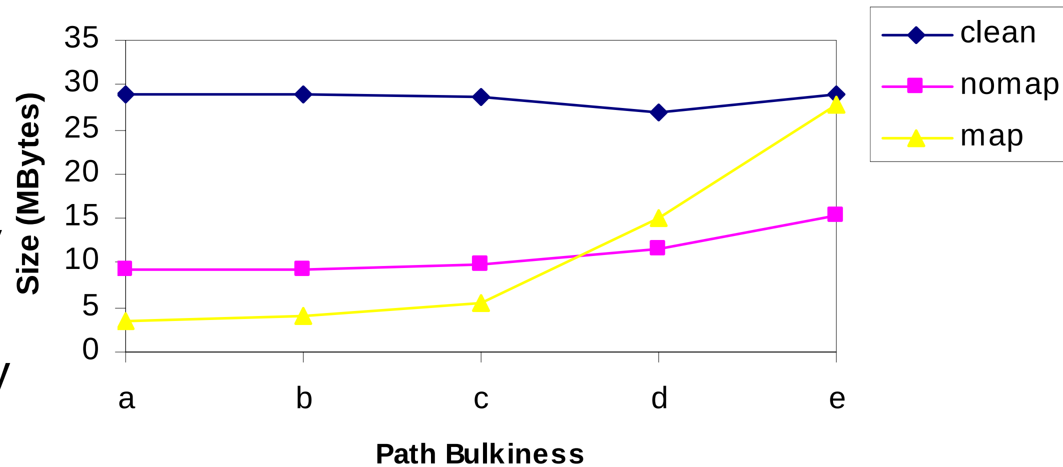


Compression vs. Data Bulkiness

P =1000, N = 1,000,000, k = 5

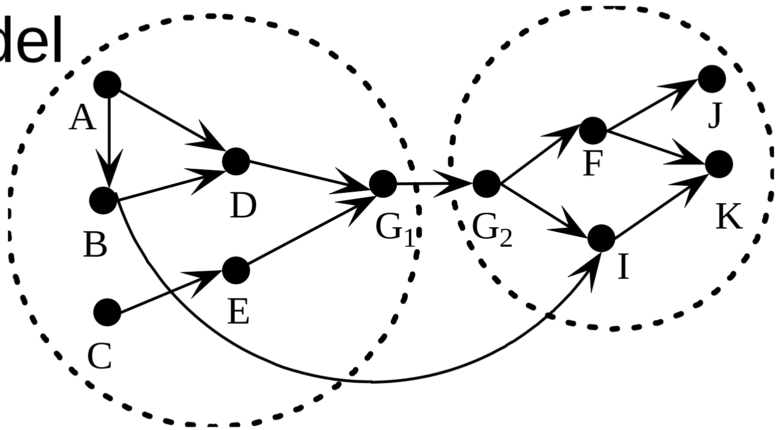
Map gives significant benefits for bulky data

For data where items move individually we are better off using tag lists

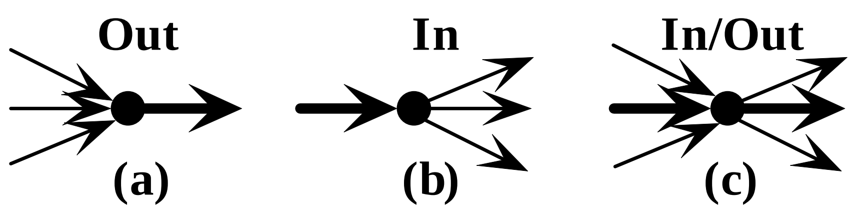


From Distribution Center Model to Gateway-Based Movement Model

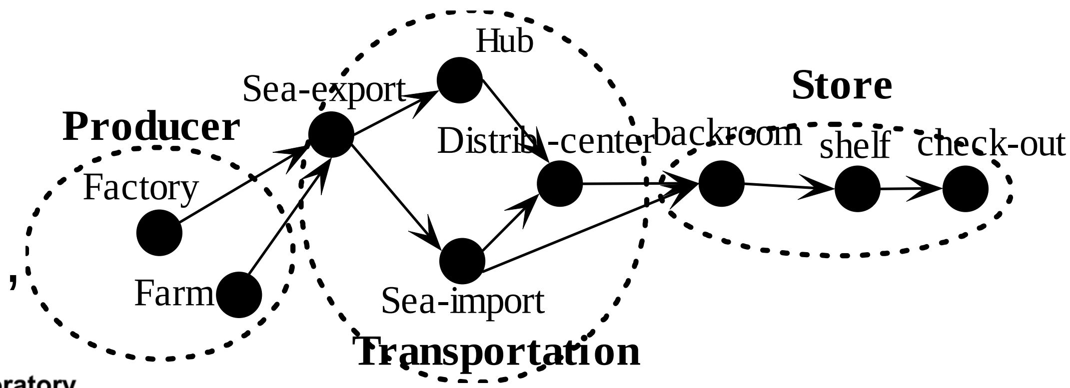
- Gateway-based movement model
  - Supply-chain movement is a merge-shuffle-split process



- Three types of gateways
  - Out, In, In/Out



- Multiple hierarchies for compression and exploration
  - Location, Time, Path, Gateway, Info



Outline

- Introduction to RFID Technology
- Why RFID Data Warehousing and Mining?
- RFID Data Warehousing
- Mining RFID Data Sets
- Conclusions

Mining RFID Data Sets

- Data cleaning by data mining
- RFID data flow analysis
- Path-based classification and cluster analysis
- Frequent pattern and sequential pattern analysis
- Outlier analysis in RFID data
- Linking RFID data mining with others

Data Cleaning by Data Mining

- RFID data warehouse substantially compresses the RFID data and facilitate efficient and systematic data analysis
- Data cleaning is essential to RFID applications
  - Multiple reading, miss reading, errors in reading, etc.
- How RFID warehouse facilitates data cleaning?
  - Multiple reading: automatically resolved when being compressed
  - Miss reading: gaps can be stitched by simple look-around
  - Error reading: use future positions to resolve discrepancies
- Data mining helps data cleaning
  - Multiple cleaning methods can be cross-validated
  - Cost-sensitive method selection by data mining

Mining RFID Data Sets

- Data cleaning by data mining
- RFID data flow analysis
- Path-based classification and cluster analysis
- Frequent pattern and sequential pattern analysis
- Outlier analysis in RFID data
- Linking RFID data mining with others

RFID Data: A Path Database View

- From raw tuples to cleansed data: A Stay Table view
  - Raw tuples: <EPC, location, time>
  - Stay view: (EPC, Location, time_in, time_out)
- A data flow view of RFID data: path forms:
  - <EPC, $(l_1,t_1),(l_2,t_2),\ldots,(l_k,t_k)$>, where l_i: location i, t_i: duration i
- The paths can be augmented with path-independent dimensions to get a Path Database of the form:
  - <Product, Manufacturer, Price, Color, $(l_1,t_1), \ldots, (l_k,t_k)$>

Path independent dimensions

Path stages

Data Flow Analysis: FlowGraph

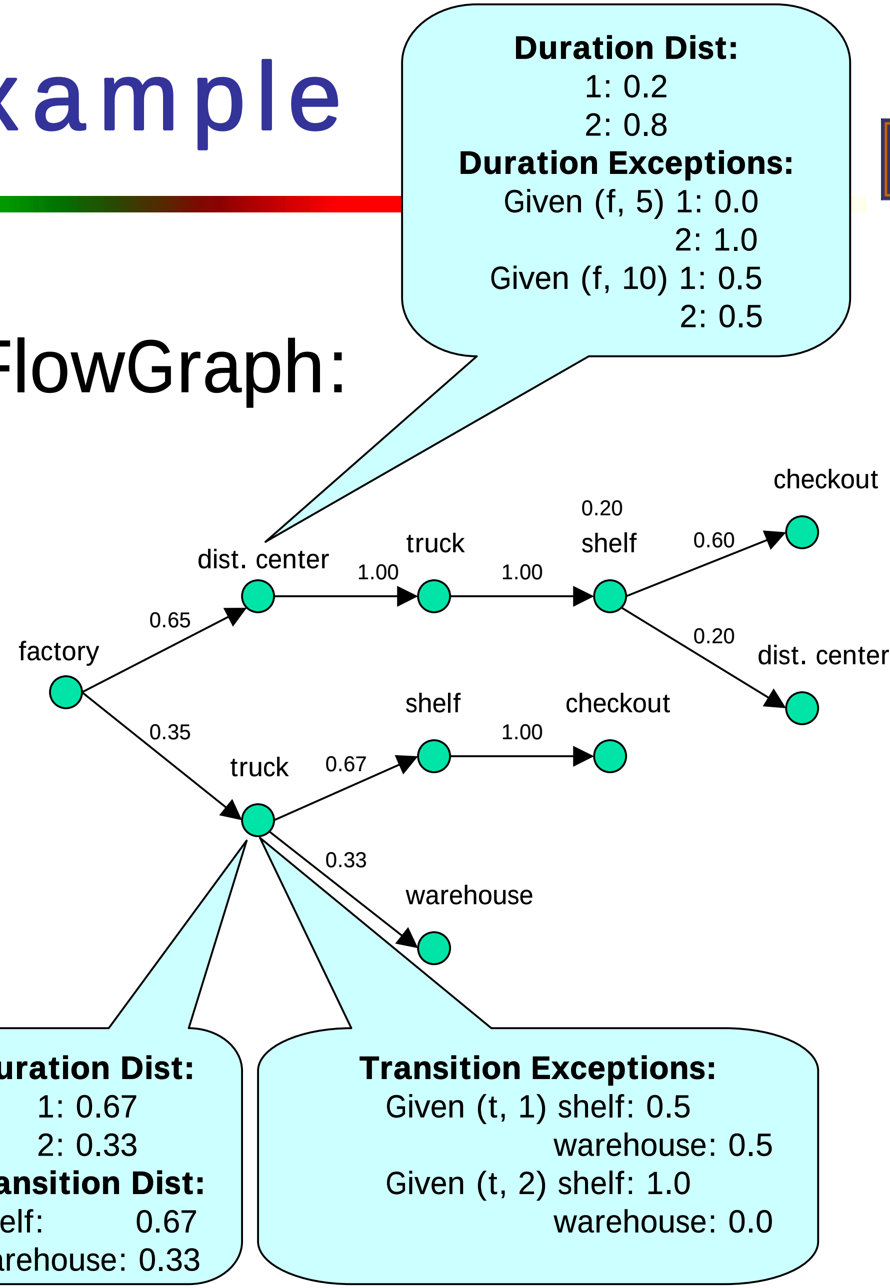
- Tree shaped workflow that summarizes the flow patterns for an item or group of items
  - Nodes: Locations
  - Edges: Transitions
- Each node is annotated with:
  - Distribution of durations at the node
  - Distribution of transition probabilities
  - Exceptions to duration and transition probabilities
    - Minimum support: frequent exceptions
    - Minimum deviation: Exceptions that have significant deviations in probability

FlowGraph: An Example

Path Database:

id	product	brand	path
1	tennis	nike	(f,10) (d,2) (t,1) (s,5) (c,0)
2	tennis	nike	(f,5) (d,2) (t,1) (s,10) (c,0)
3	sandals	nike	(f,10) (d,1) (t,2) (s,5) (c,0)
4	shirt	nike	(f,10) (t,1) (s,5) (c,0)
5	jacket	nike	(f,10) (t,2) (s,5) (c,1)
6	jacket	nike	(f,10) (t,1) (w,5)
7	tennis	adidas	(f,5) (d,2) (t,2) (s,20)
8	tennis	adidas	(f,5) (d,2) (t,3) (s,10) (d,5)

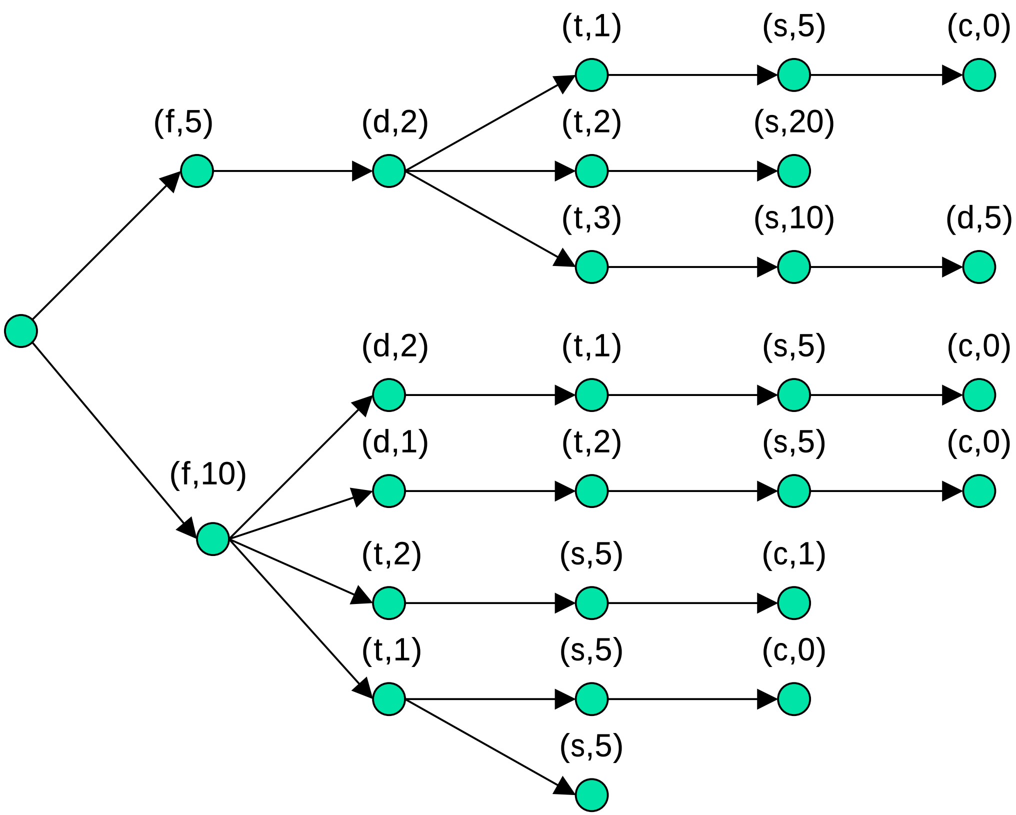
FlowGraph:



FlowGraph: Alternative Design

- The FlowGraph incorporates duration information in a compact manner
- If we create nodes for each distinct path stage the size of the workflow may explode
- Duration-dependent nodes may add little information when transition and duration probabilities are largely path independent

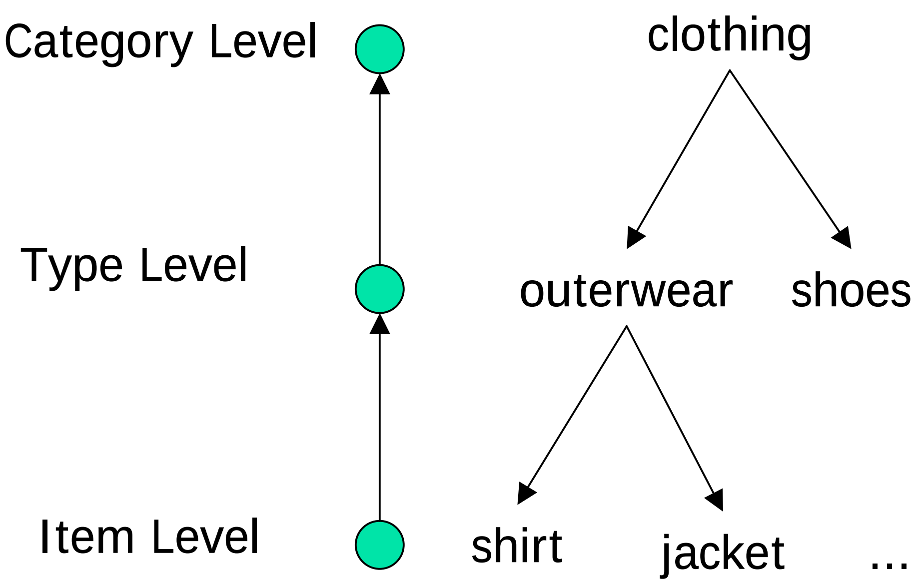
Duration-Dependent Nodes



Item Abstraction Level

- Each path independent dimension has an associated concept hierarchy
- The set of concept hierarchies for all path independent dimensions forms an item lattice.
- The path independent dimensions can be aggregated to a given level in the item lattice.

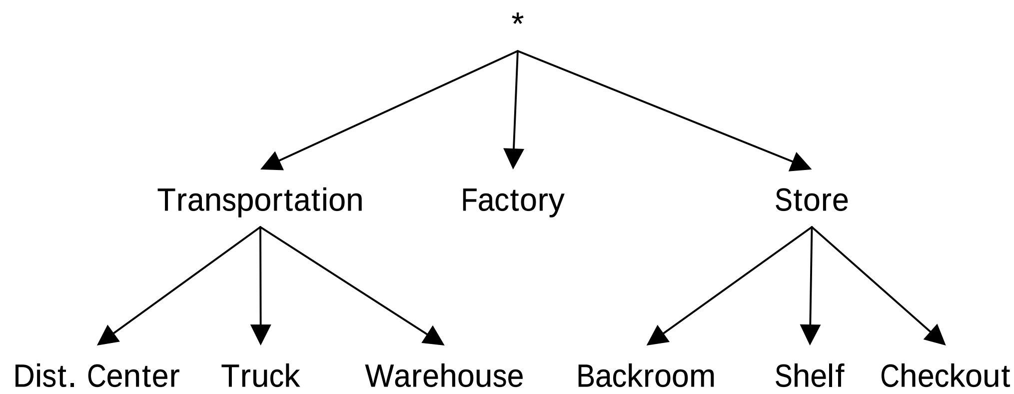
Product Concept Hierarchy



Path Abstraction Level

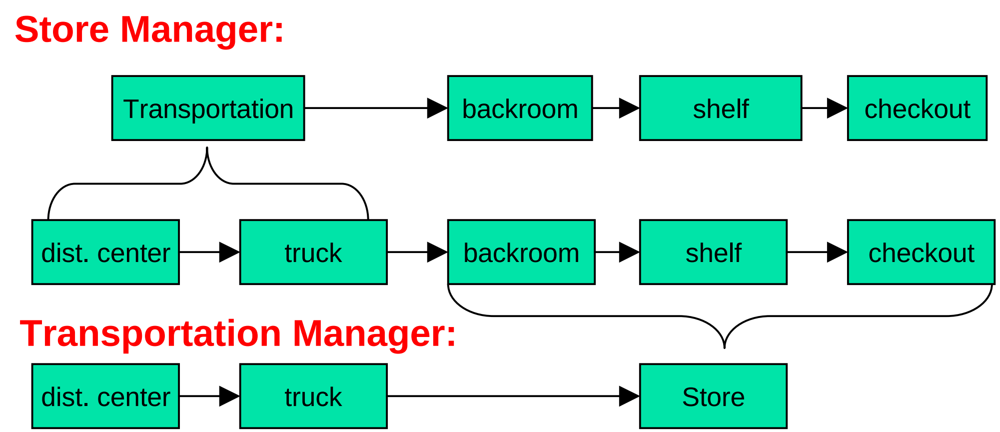
Location Lattice

- The levels for the location and time dimensions of each path stage form a path lattice.
- Path stages can be aggregated to a given level in the path lattice.



Path Views

- Each path can be aggregated at different abstraction levels
- We collapse path stages along the location concept hierarchy



FlowCube

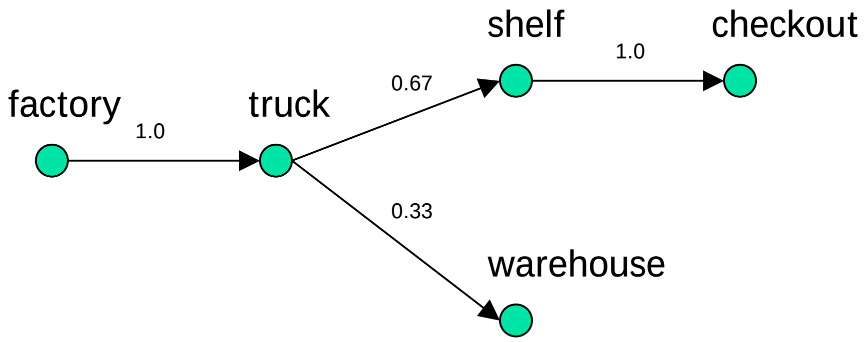
- Data cube computed on the path database, by grouping entries that share the same values on the path independent dimensions.
- Each cuboid has an associated level in the item and path abstraction lattices.
  - Level in the item lattice.
    - (product category, country, price)
  - Level in the path lattice.
    - (<transportation, factory, backroom, shelf, checkout>, hour)
- The measure for each cell in the FlowCube is a FlowGraph computed on the paths aggregated in the cell.

FlowCube Example

Cuboid for <product type, brand>

cell id	product	brand	path ids
1	shoes	nike	1,2,3
2	shoes	adidas	7,8
3	outerwear	nike	4,5,6

FlowGraph for cell 3



Mining RFID Data Sets

- Data cleaning by data mining
- RFID data flow analysis
- Path-based classification and cluster analysis
- Frequent pattern and sequential pattern analysis
- Outlier analysis in RFID data
- Linking RFID data mining with others

Path- or Segment- Based Classification and Cluster Analysis

- Classification: Given class label (e.g., broken goods vs. quality ones), construct path-related predictive models
  - Take paths or segments as motifs and perform motif-based high-dimensional information for classification
- Clustering: Group similar paths or similar stay or movements of RFIDs, with other multi-dimensional information into clusters
  - It is essential to define new distance measure and constraints for effective clustering

Mining RFID Data Sets

- Data cleaning by data mining
- RFID data flow analysis
- Path-based classification and cluster analysis
- Frequent pattern and sequential pattern analysis
- Outlier analysis in RFID data
- Linking RFID data mining with others

Frequent Pattern and Sequential Pattern Analysis

- Frequent patterns and sequential patterns can be related to movement segments and paths
- Taking movement segments and paths base units, one can perform multi-dimensional frequent pattern and sequential pattern analysis
- Correlation analysis can be formed in a similar way
  - Correlation components can be stay, move segments, and paths
- Efficient and scalable algorithms can be developed using the warehouse modeling

Mining RFID Data Sets

- Data cleaning by data mining
- RFID data flow analysis
- Path-based classification and cluster analysis
- Frequent pattern and sequential pattern analysis
- Outlier analysis in RFID data
- Linking RFID data mining with others

Outlier Analysis in RFID Data

- Outlier detection in RFID data is by-product of other mining tasks
  - Data flow analysis: Detect those not in the major flows
  - Classification: Treat outliers and normal data as different class labels
  - Cluster analysis: Identify those that are deviate substantially in major clusters
  - Trend analysis: Those not following the major trend
  - Frequent pattern and sequential pattern analysis: anomaly patterns

Mining RFID Data Sets

- Data cleaning by data mining
- RFID data flow analysis
- Path-based classification and cluster analysis
- Frequent pattern and sequential pattern analysis
- Outlier analysis in RFID data
- Linking RFID data mining with others

Linking RFID Mining with Others

- RFID warehouse and cube model makes the data mining better organized and more efficient
- Real time RFID data mining will need further development of stream data mining methods
  - Stream cubing and high dimensional OLAP are two key method that will benefit RFID mining
- RFID data mining is still a young, largely unexplored field
- RFID data mining has close links with sensor data mining, moving object data mining and stream data mining
  - Thus will benefit from rich studies in those fields

Outline

- Introduction to RFID Technology
- Why RFID Data Warehousing and Mining?
- RFID Data Warehousing
- Mining RFID Data Sets
- Conclusions

Conclusions

- A new RFID warehouse model
  - Allows efficient and flexible analysis of RFID data in multidimensional space
  - Preserves the structure of the data
  - Compresses data by exploiting bulky movements, concept hierarchies, and path collapsing
- Mining RFID data
  - Powerful mining mechanisms can be constructed with RFID data warehouse
  - Flowgraph analysis, data cleaning, classification, clustering, trend analysis, frequent/sequential pattern analysis, outlier analysis
- Lots can be done in RFID data analysis

References of the Talk

- Hector Gonzalez, Jiawei Han, Xiaolei Li, and Diego Klabjan, "Warehousing and Analysis of Massive RFID Data Sets", in Proc. 2006 Int. Conf. on Data Engineering (ICDE'06), Atlanta, Georgia, April 2006.
- Hector Gonzalez, Jiawei Han, and Xiaolei Li, "FlowCube: Constructuing RFID FlowCubes for Multi-Dimensional Analysis of Commodity Flows", in Proc. 2006 Int. Conf. on Very Large Data Bases (VLDB'06), Seoul, Korea, Sept. 2006.
- Hector Gonzalez, Jiawei Han, and Xiaolei Li, "Mining Compressed Commodity Workflows From Massive RFID Data Sets", in Proc. 2006 Int. Conf. on Information and Knowledge Management (CIKM'06), Arlington, VA, Nov. 2006.
- Hector Gonzalez, Jiawei Han, and Xuehua Shen, "Cost-conscious Cleaning of Massive RFID Data Sets", Proc. 2007 Int. Conf. on Data Engineering (ICDE'07), Istanbul, Turkey, April 2007.

Thanks and Questions