Integration and Automation of Data Preparation and Data Mining

Yanhui Geng
Huawai Technologies

Agenda

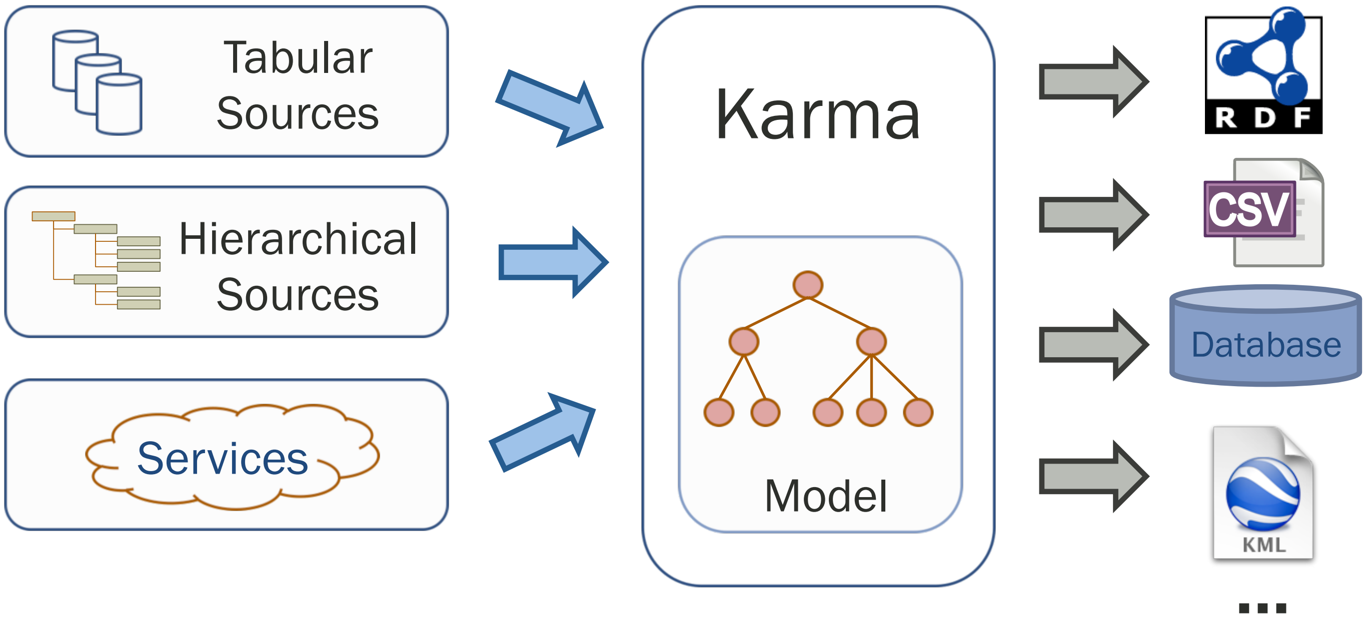
- Introduction
- Karma – Data Modeling and Integration
- Prediction Task
- Data collection
- Preparing the mode of transportation data
- Using Karma
- Our Approach - Karma Workflow
- Evaluation
- Related Work
- Discussion

Introduction

- Data preparation – To transform the raw data into a form that could be consumed by mining tools
- Raw data collected is heterogeneous, noisy, inconsistent and incomplete
- Data Preparation is an iterative task
- Preparation tasks - cleaning, discretization, transformation and data integration
- Consumes 70 to 80% of the total time

Karma

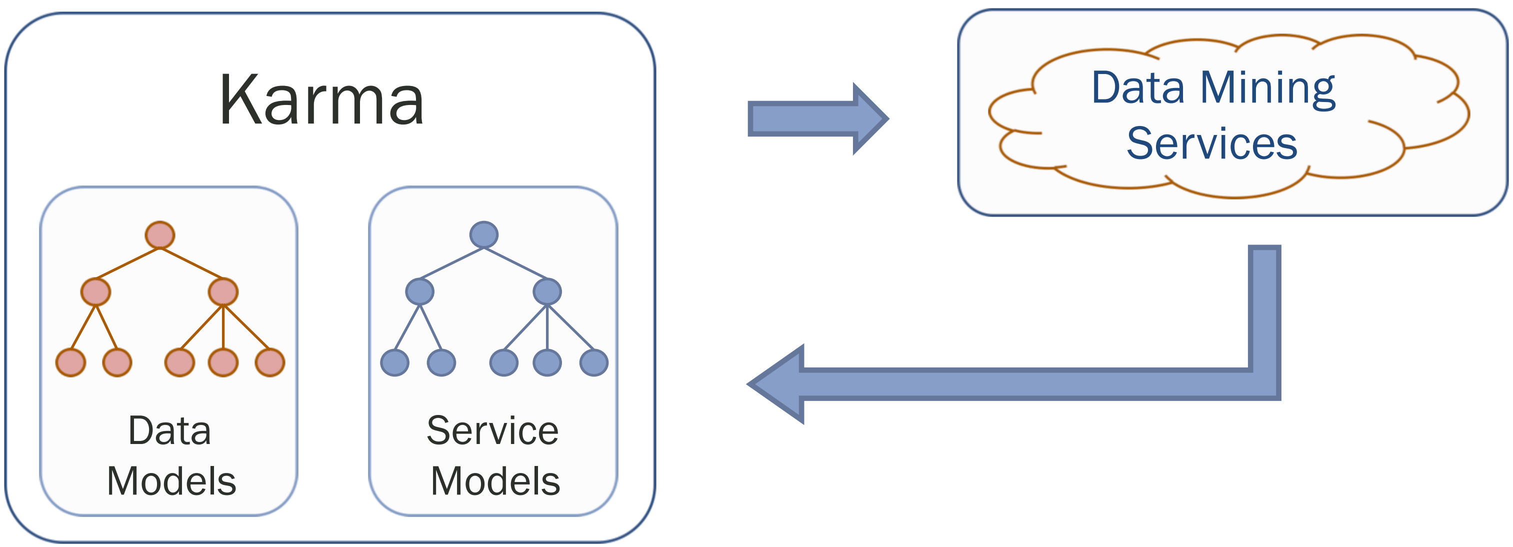
Interactive tool for rapidly extracting, cleaning, transforming, and publishing data



[Knoblock, Szekely, et al. Semi-automatically mapping structured sources into the semantic web. ISWC 2012]

Karma cont’d

We propose to combine the steps in data preparation and data mining into a single integrated process using Karma



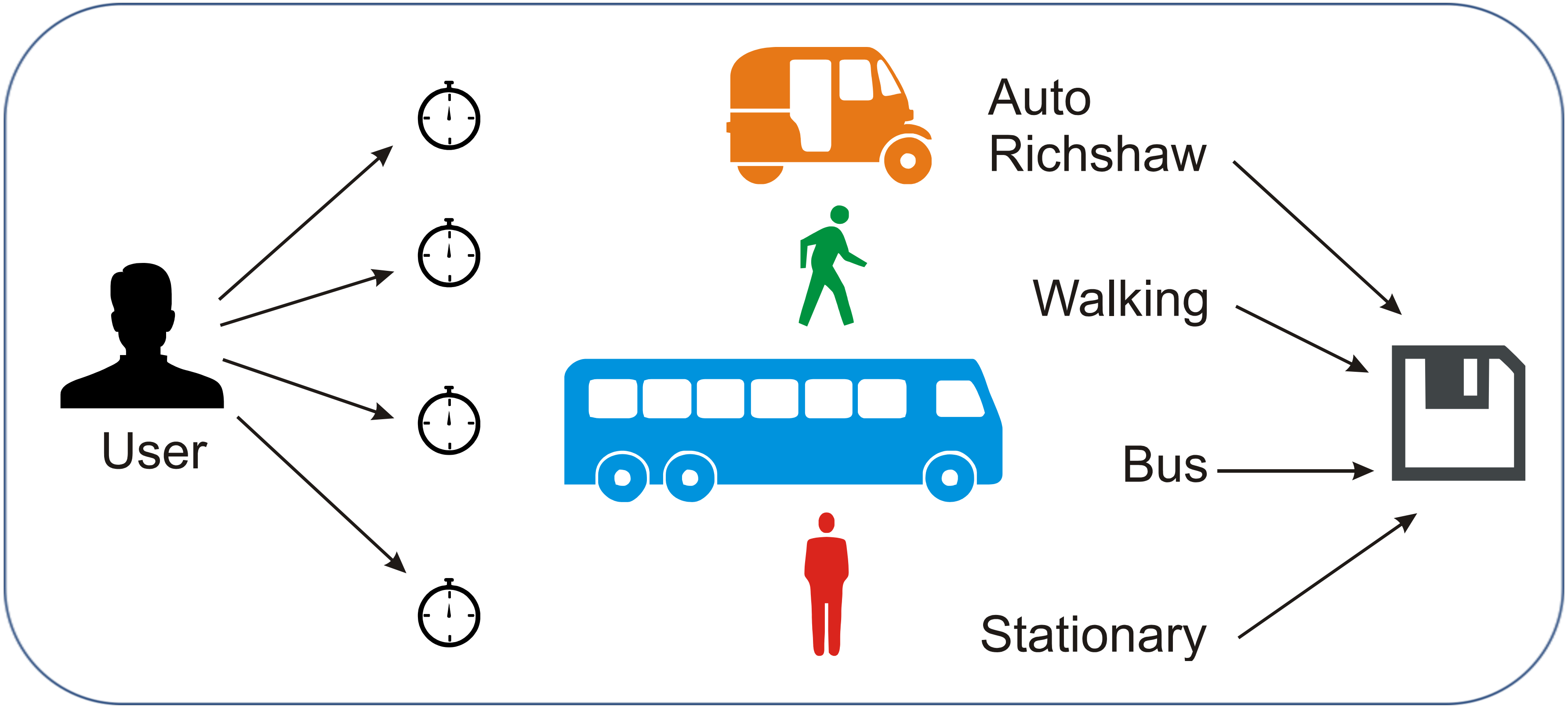
Capture detailed metadata about the data sources, transformations and mining services that are invoked.

Predicting the Mode of transportation

- Collect data from GPS and Accelerometer sensors
- Record mode of transport labels
- Extract and transform collected data to generate useful features
- Split the dataset into training and testing sets
- Use Support Vector Machine (SVM) algorithm to train a model with the training data
- Predict mode of transport on records in the testing data

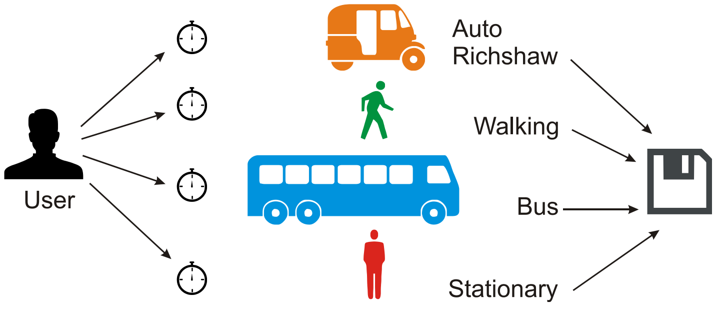
Data Collection

Collected Accelerometer and GPS sensor data using Android App for different modes of transportation

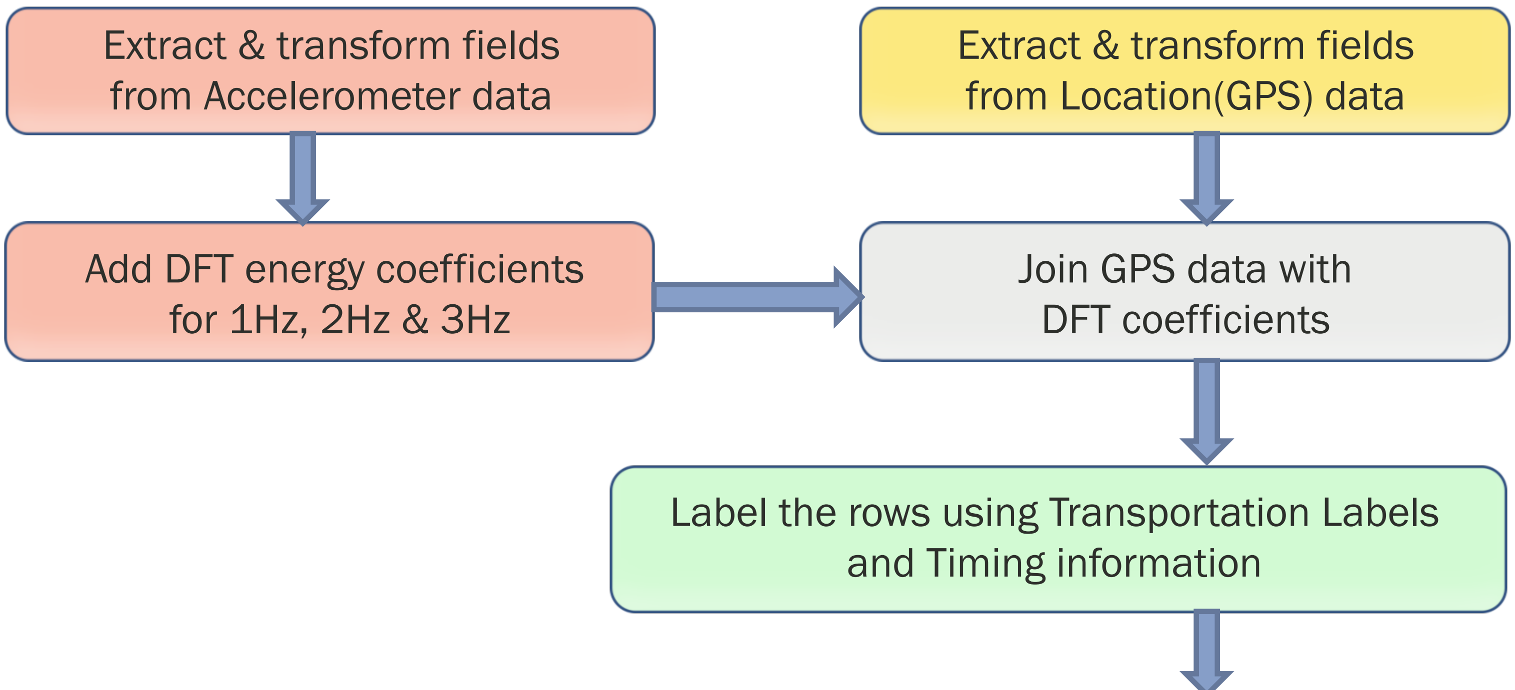


Data collection cont'd

- Total 3 days data was collected
- For each day we have 3 csv files
  - *AccelerometerSensor.csv*
  - *LocationProbe.csv*
  - *TransportationLabels.csv*
- User manually noted the time period for each mode of transportation used

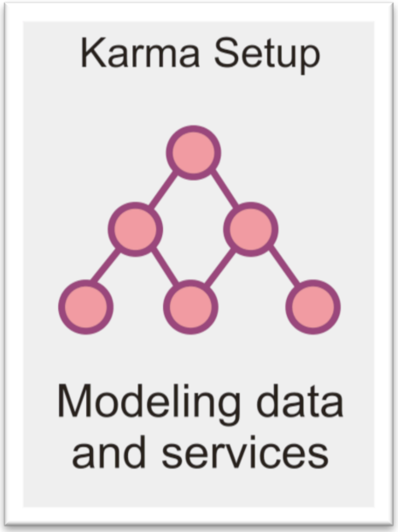


Preparing the mode of transportation data



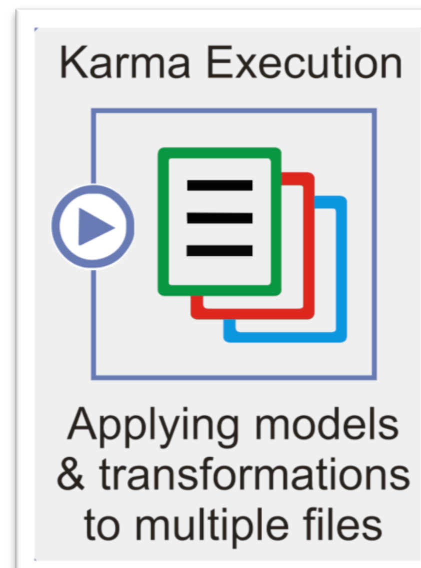
timestamp	speed	accuracy	acceleration magnitude	DFT_E1	DFT_E2	DFT_E3	mode
1387869469	0	16	11.69130897	136.686705	139.957767	139.957767	walking
1388062990	0.89422005	8	11.8207537	139.730218	139.730218	135.891275	stationary
1388060907	2.3307722	12	12.17176955	148.151974	148.151974	146.537468	bus
1388059088	7.702458	12	14.09193116	198.582524	92.5838217	104.223227	auto

Using Karma

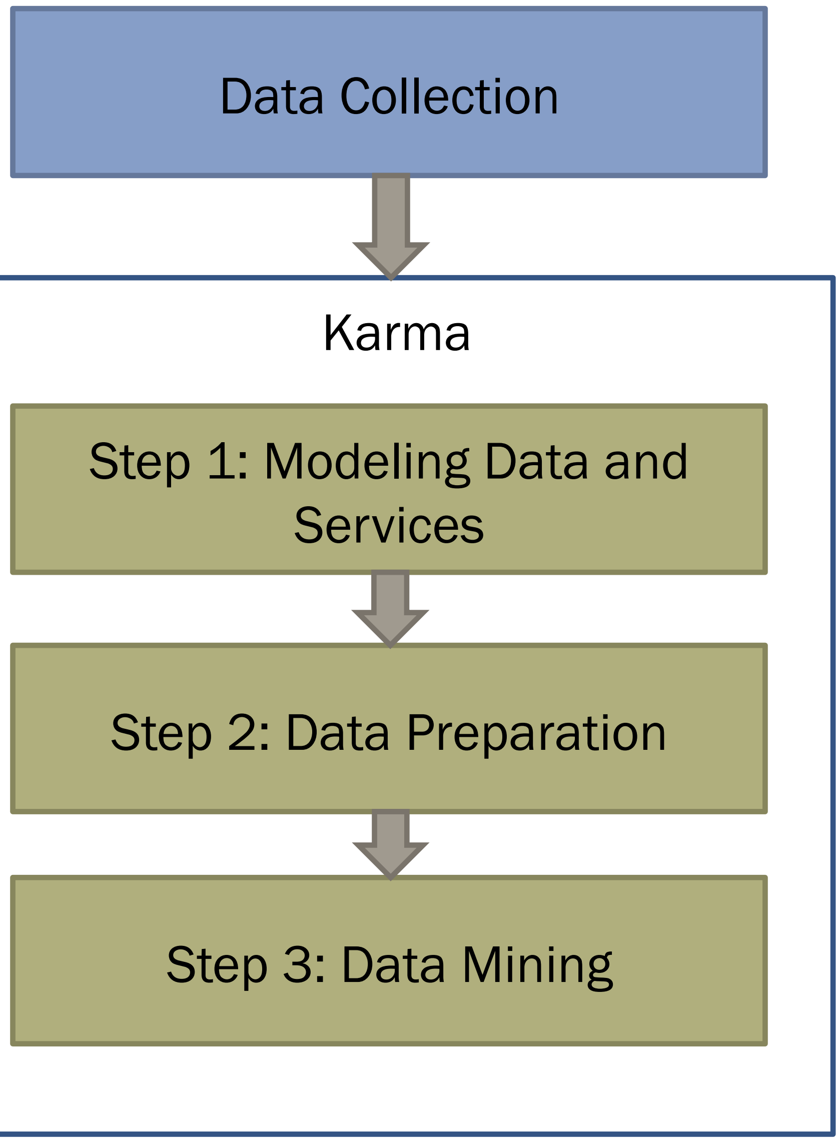


- These tasks are performed only the first time
- Modeling the raw datasets and the required web services
- All transformations and processing done here is recorded by Karma

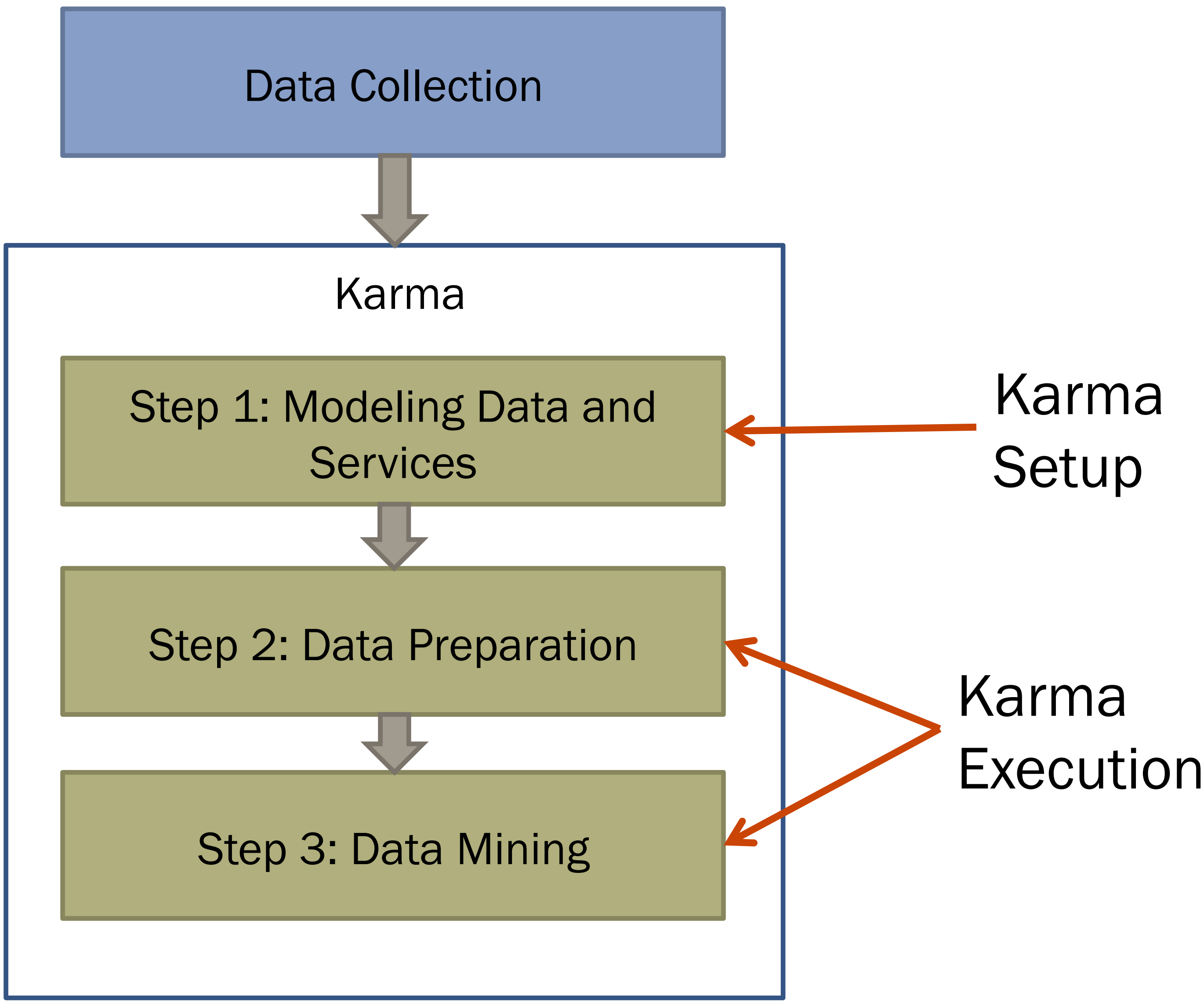
- The Karma execution tasks are ones that are repeated for each dataset.
- Applying transformations, join operations and invoking the data mining services



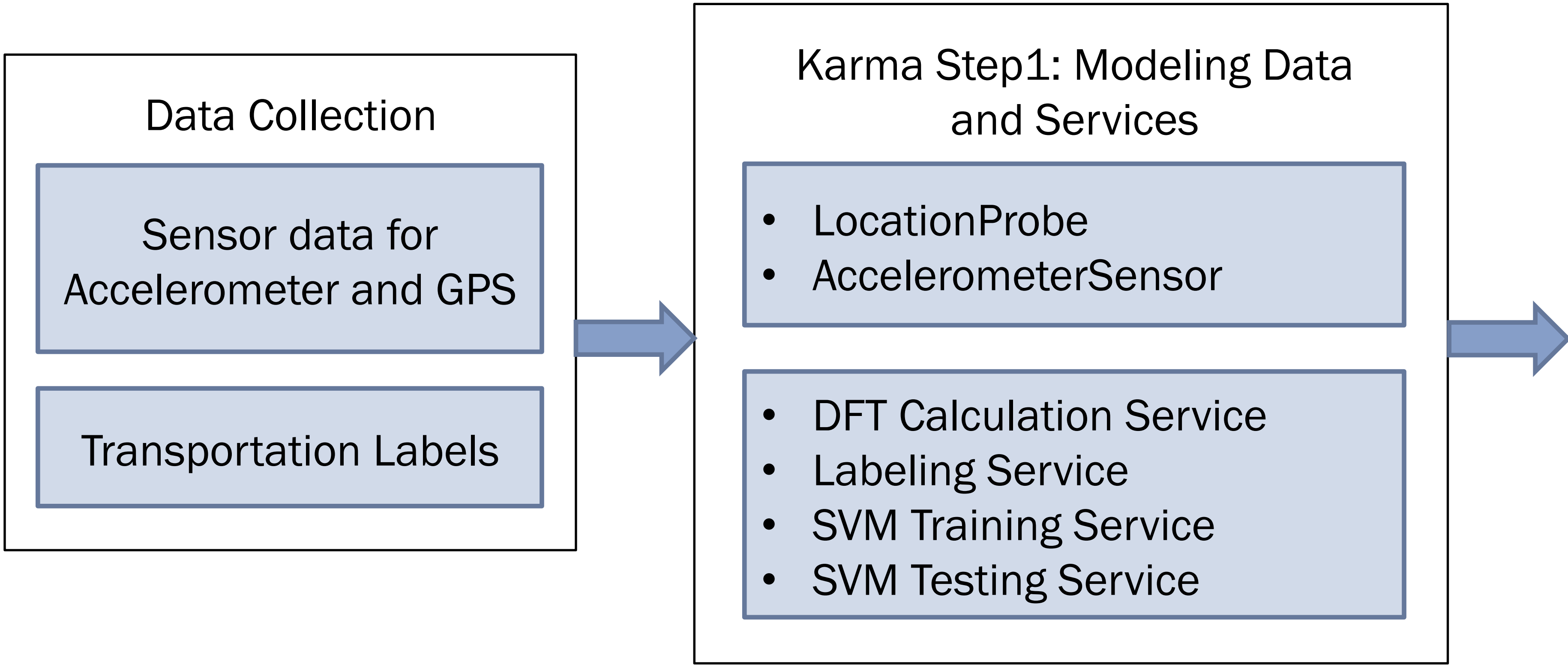
Workflow using Karma



Workflow using Karma



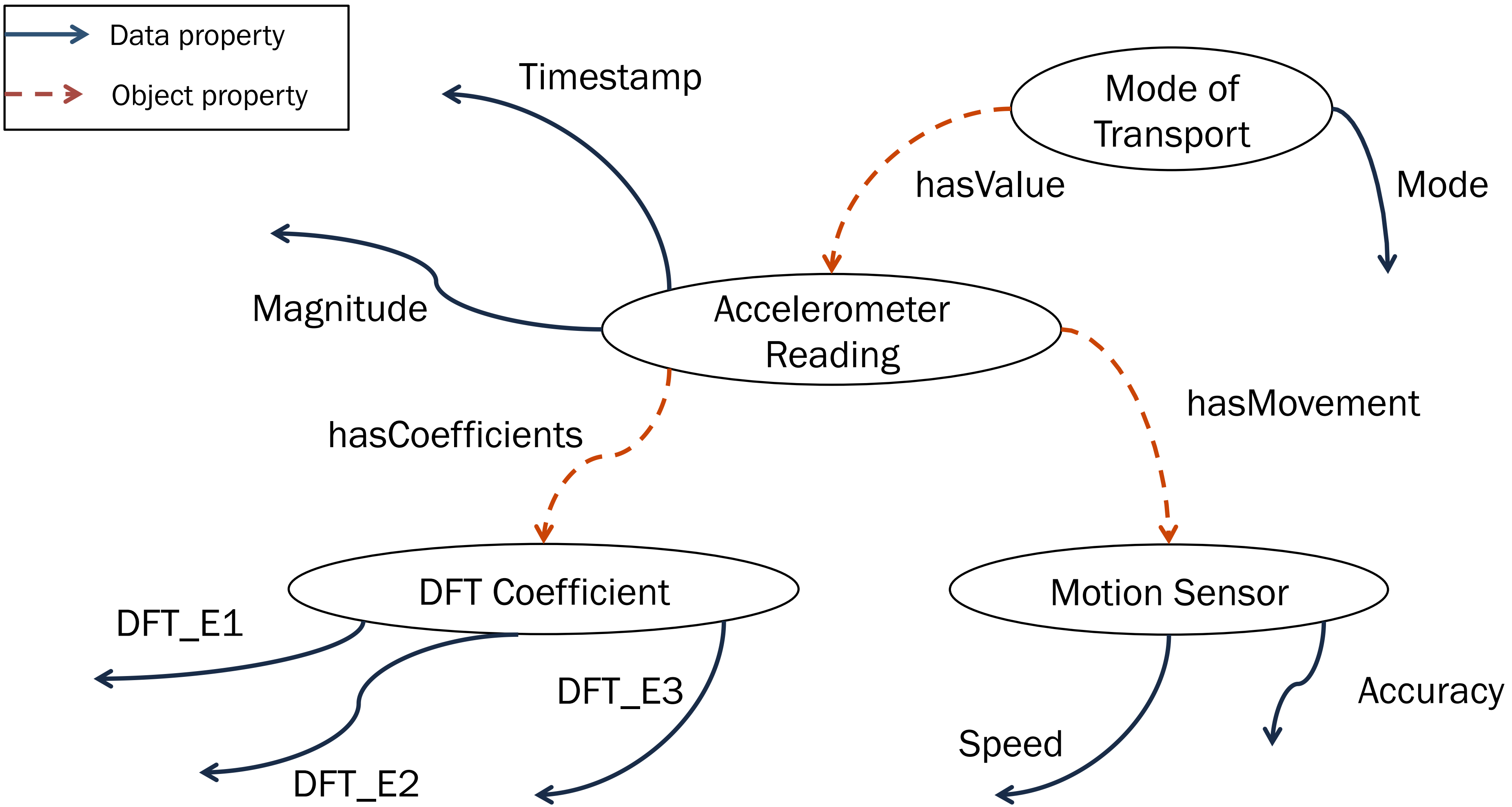
Workflow cont'd



Workflow cont'd

Karma Step 1: Modeling Data and Services

Applying a Semantic Model to the data set

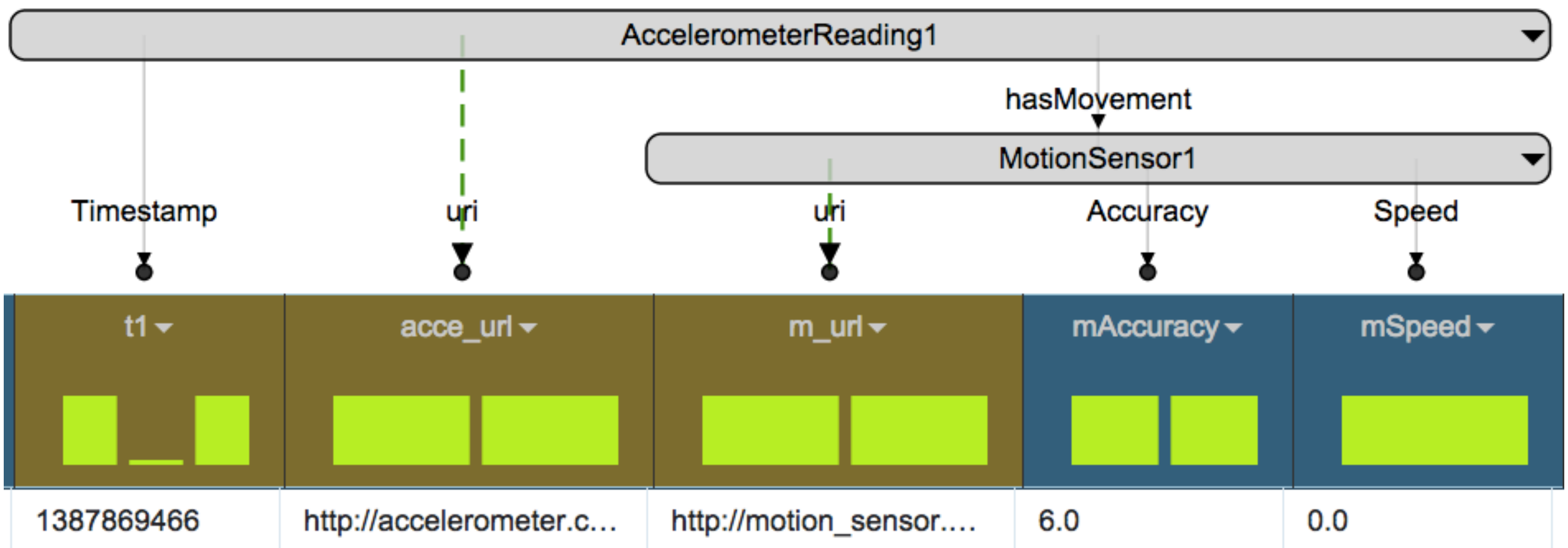


Workflow cont'd

Karma Step 1: Modeling Data and Services

Modeling the LocationSensor Data

- Round off the timestamp column using Python transform
- We model only the required columns - timestamp, accuracy and speed and add URLs for both the classes using the timestamp values
- Publish the RDF

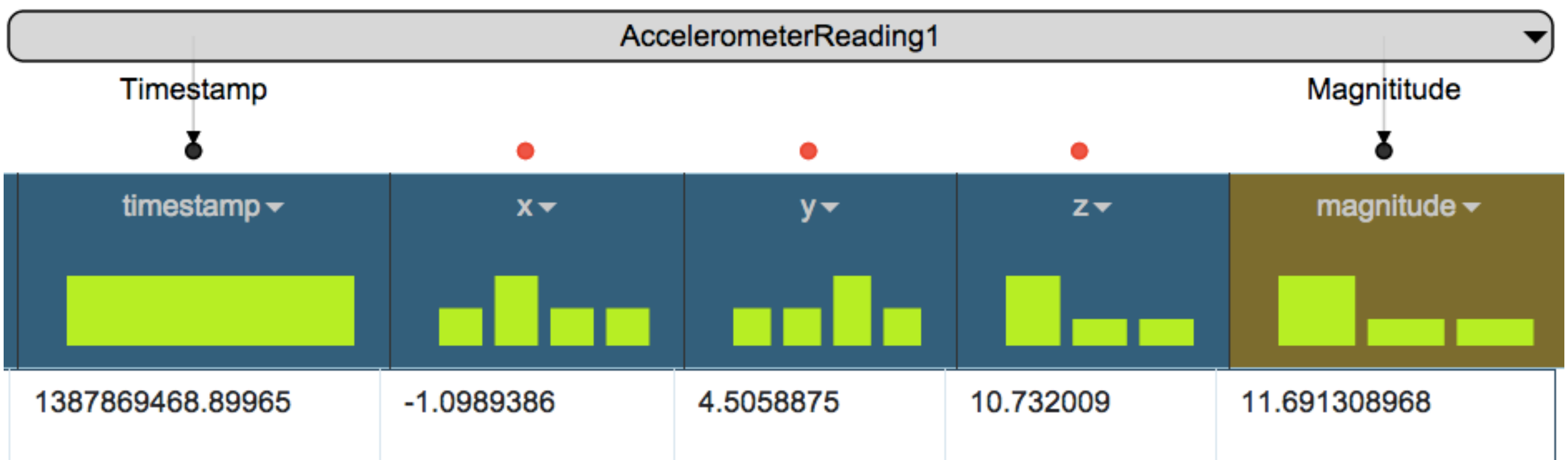


Workflow cont'd

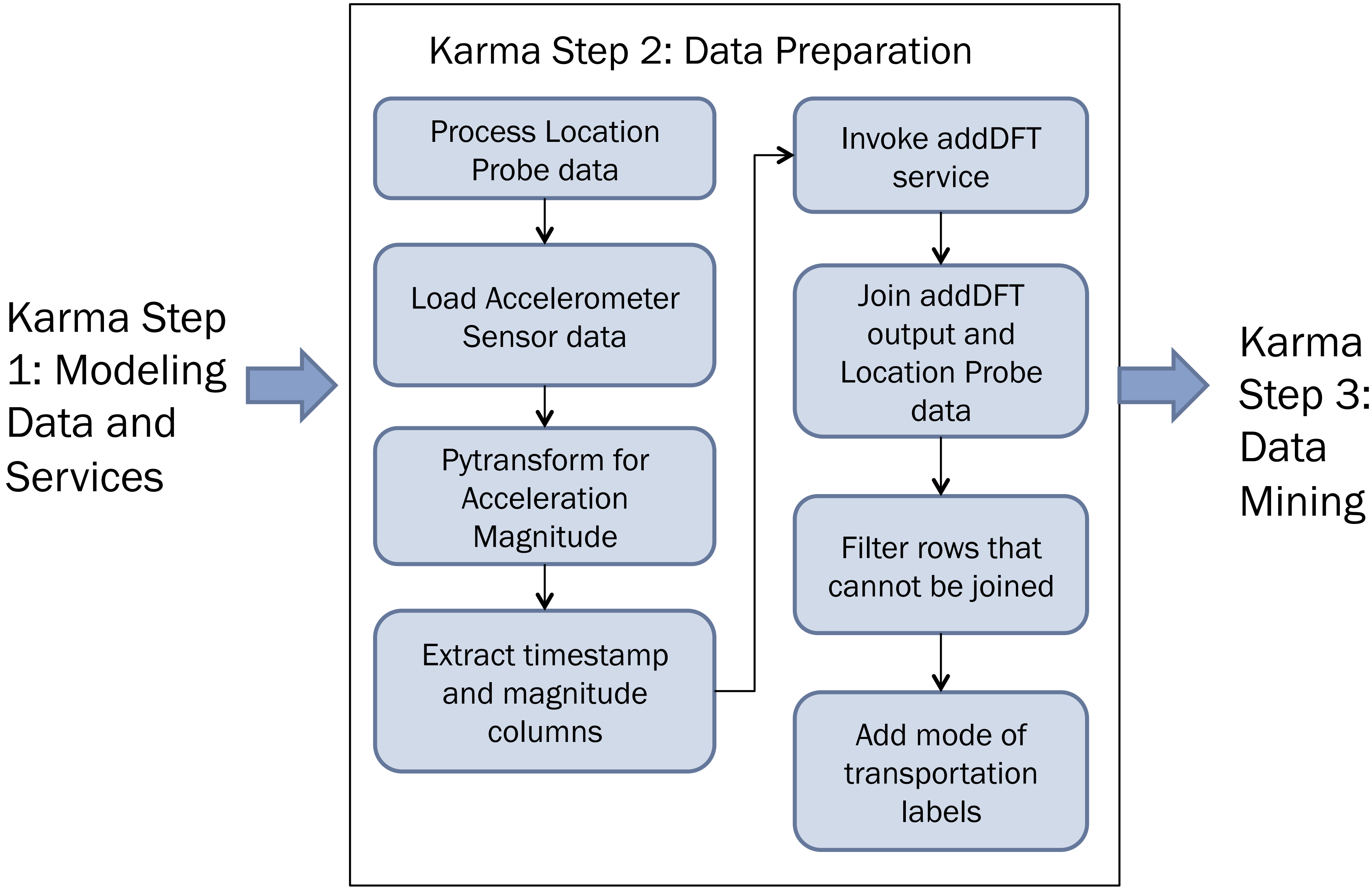
Karma Step 1: Modeling Data and Services

Modeling the DFT service

- Calculate "Magnitude" using a Python transformation as *magnitude = sqrt(*$x^2 + y^2 + z^2$*)*
- Set semantics for the timestamp and magnitude columns
- Set additional properties like service url, method, etc. and publish the model



Workflow cont'd



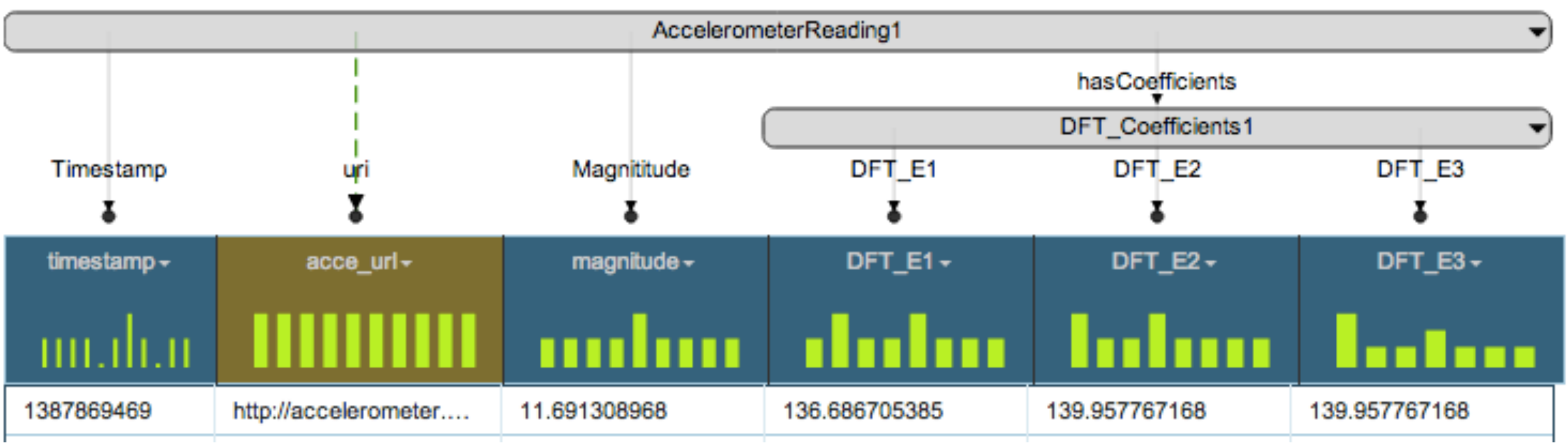
Workflow cont'd

Karma Step 2: Data Preparation

Processing Accelerometer files

- Apply the 'AccelerometerSensor' model and publish the data

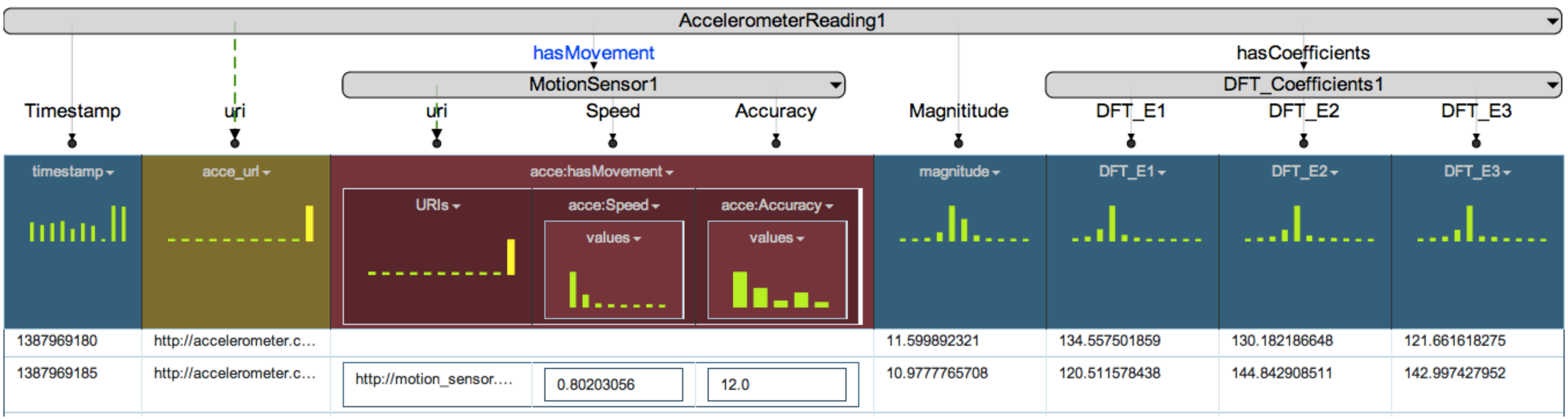
- Invoke the DFT service. The DFT service produces a new worksheet which contains the new columns for DFT coefficients



Workflow cont'd

Karma Step 2: Data Preparation

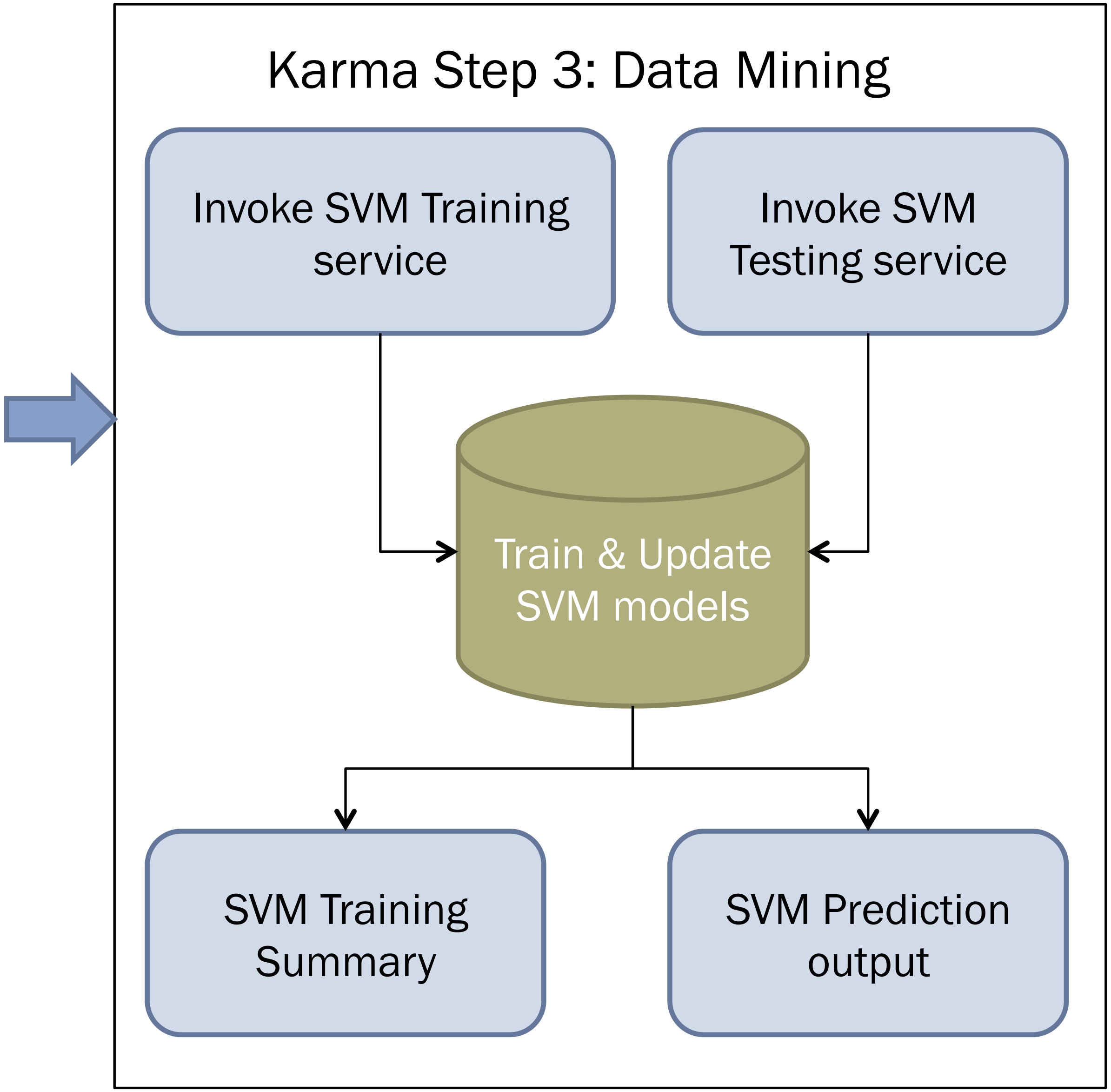
- Add the url for 'AccelerometerReading' class
- Publish the data
- Join the data with the location dataset



- Invoke the labeling service on the augmented dataset

Workflow cont'd

Karma Step 2:
Data
Preparation



Workflow cont’d

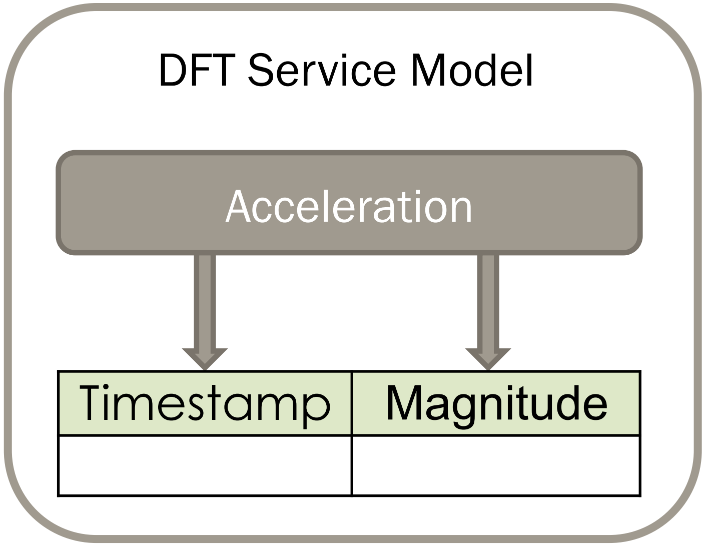
Karma Step 3: Data Mining

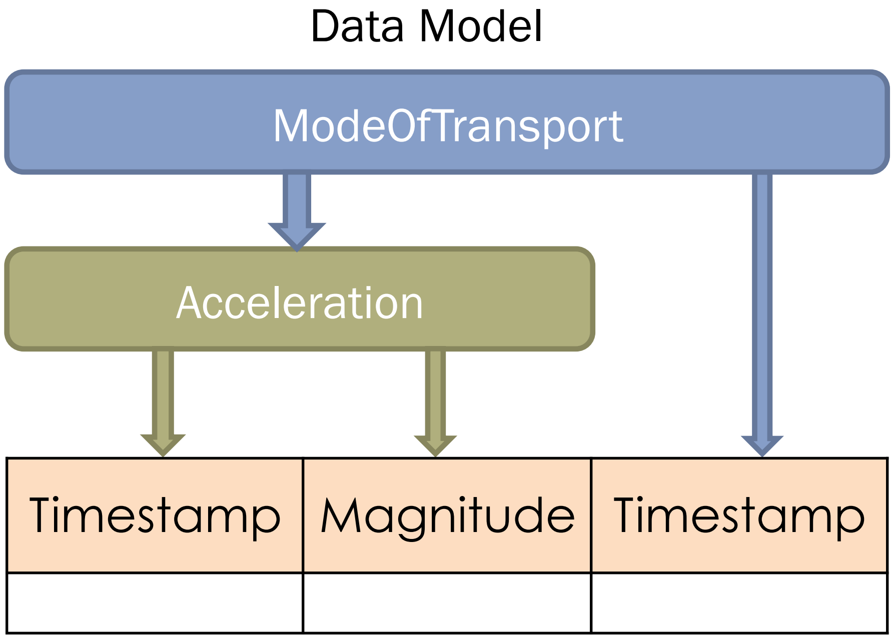
- Karma automatically identifies which services can be invoked on the current data
- Karma matches the semantic types and the relationship between the classes of the data with all the service models in the repository
- A list of services is shown to the user along with the number of properties it uses as inputs for the service

Workflow cont'd

Karma Step 3: Data Mining

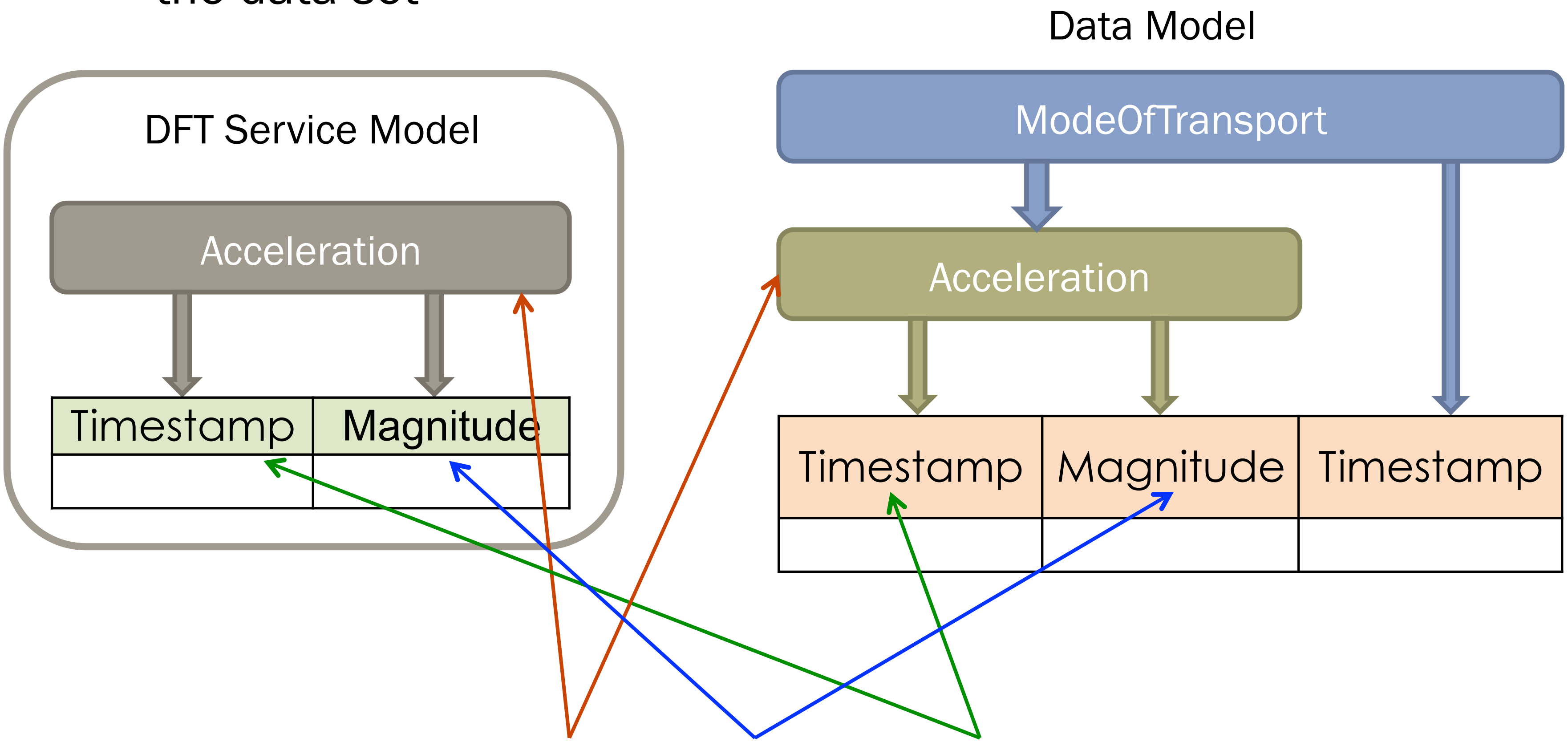
How Karma identifies services that could be invoked on the data set





Workflow cont'd
Karma Step 3: Data Mining

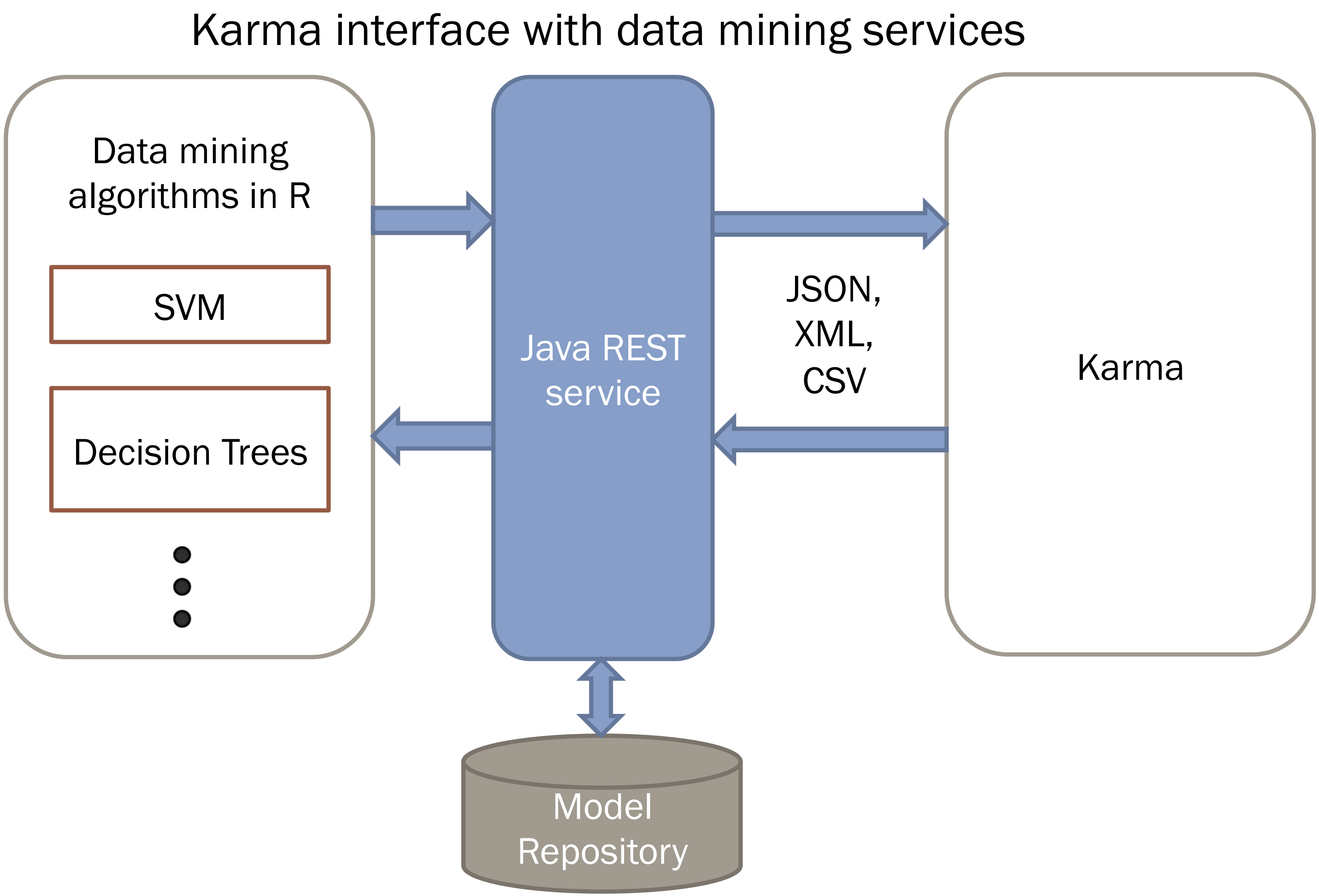
How Karma identifies services that could be invoked on the data set



Karma matches the class and semantic types and determines that the DFT service can be invoked

Workflow cont'd

Karma Step 3: Data Mining



Workflow cont'd

Karma Step 3: Data Mining

- Karma can interact with a web service using the service model
- In our current example, the SVM is implemented in R programming language
- A Java based REST service is used as an interface for the R programs
- The REST service keeps tracks of all the models that were trained using a unique model identifier

Evaluation

- We evaluated our approach by measuring
  - Reduction in the time and
  - Reduction in effort required to perform data preparation and data mining for the mode of transport prediction task
- We compared the time taken using Karma and MS Excel
- The effort and time to write scripts for DFT calculation, SVM, etc. were excluded as they were part of both approaches

Evaluation cont'd

Using MS Excel

1. Merge the LocationProbe.csv file from each day into a single file
2. Processing AccelerometerSensor.csv
    1. Transform Timestamp column
    2. Calculate Magnitude for each row in a new column
    3. Save in a new file
3. Invoke python script for DFT calculations on the previous file
4. Processing LocationProbe.csv
    1. Extract Timestamp, Accuray and Speed columns in a new sheet
    2. Transform Timestamp column
    3. Join the output of DFT calculation script with the LocationProbe file to attach Speed and Accuracy columns.
    4. Save the file
5. Invoke the python script for labeling the joined data
6. Invoke the SVM training script

Evaluation cont'd

Time taken by Karma for one trial of data processing and data mining

Step	Task	User Time (sec)	System Processing Time (sec)	Total Elapsed Time
1	Modeling LocationProbe data	34	18	0:52
2	Publish RDF for LocationProbe	12	6	1:10
3	Modeling AccelerometerSensor data	18	5	1:34
4	Publish RDF for AccelerometerSensor	11	9	1:54
5	Invoke addDFT service	8	2	2:04
6	Modeling DFT service output	10	2	2:16
7	Publish RDF for DFT output	11	6	2:33
8	Join with LocationProbe RDF	12	5	2:50
9	Publish the augmented model	15	3	3:08
10	Publish RDF for joined data	10	6	3:24
11	Invoke getLabel service	8	2	3:34
12	Filter our 'NA' mode of transport	31	3	4:08
12	Model mode of transport data - the result of add label service	6	3	4:17
13	Publish RDF for Model of transport data	20	4	4:41

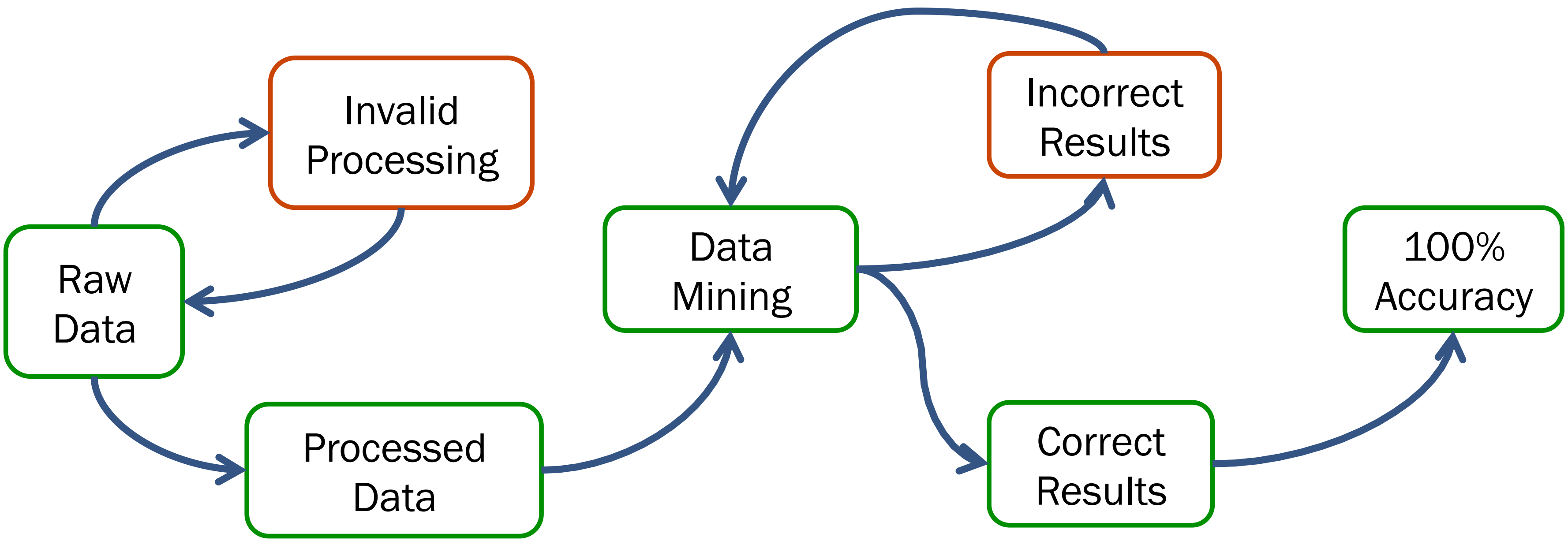
Evaluation cont'd

Time taken by MS Excel for one trial of data processing and data mining

Step	Task	User Time (sec)	System Processing Time (sec)	Total Elapsed Time
1	Process AccelerometerSensor data — add magnitude and set timestamp column to be 4 decimal places	44	0	0:44
2	Extract timestamp and Magnitude in new worksheet and save as CSV	41	0	1:25
3	Invoke addDFT script	8	2	1:35
4	Process addDFT output file — for- mat timestamp column to be 4 dec- imal places	12	0	1:48
5	Copy timestamp, speed and accu- racy columns from LocationProbe data into a new worksheet	41	0	2:29
6	Process timestamp column to be 4 decimal places, and add a new col- umn to round off the decimal	25	0	2:54
7	Add vLookUp formulae in the AccelerometerData worksheet for Speed	27	0	3:21
8	Add vLookUp formulae in the Ac- celerometerData worksheet for Ac- curacy	34	0	3:55
9	Apply filter to remove unmatched — NA rows after join and delete them.	43	0	4:38
10	Save this accelerometer with DFT data for input to labeling service	19	0	4:57
11	Invoke the labeling service over the exported CSV file	12	1	5:09
12	Filter data to remove NA columns	32	0	5:41
13	Save the file as ProcessedData file	6	0	5:48
14	Copy the ProcessedData file to the required location for SVM invoca- tion	10	0	5:58

Evaluation cont'd

- We performed two trials of data preparation and data mining
- Each trial consisted of 3 days data
- Accuracy is 100% for both approaches because the user can always go back and rectify an error in data preparation or data mining



Evaluation cont'd

	Karma	MS Excel
Total time for trail 1 and 2	22:39 min	40:20 min

Total Reduction excluding karma setup	17:41 min	42.14%
Total Reduction including karma setup (Set up time : 9:30 min)	8:11 min	20.28%
Accuracy with Karma		100.00%
Accuracy with Excel		100.00%

Related Work

- RapidMiner and KNIME
  - \+ have data preparation features
  - \+ have support for invoking remote web services
  - \+ offer integration of data preparation and mining
  - − lack semantic definition of remote web services that can be published and shared

- DataPreparator and Google Refine
  - \+ have data preparation features
  - − do not offer integration of data preparation and mining

- Our Work
  - offers all the above features (bulleted with '+')
  - offers semantic definition of remote web services
  - offers automation of data preparation tasks

Discussion

- An end-to-end approach of data preparation and data mining
- Ability to share models across users by using semantic web technology
- Users need not be an expert in machine learning or have advanced programming skills to perform data mining

Thank You