D1.2 Final data manager and data warehouse storage infrastructure

DATA science for SIMulating the era of electric vehicles

www.datasim-fp7.eu

Project details	**Project reference:** 270833 **Status:** Execution **Programme acronym:** FP7-ICT (FET Open) **Subprogramme area:** ICT-2009.8.0 Future and Emerging Technologies **Contract type:** Collaborative project (generic)
Consortium details	**Coordinator:** 1. (UHasselt) Universiteit Hasselt **Partners:** 2. (CNR) Consiglio Nazionale delle Ricerche 3. (BME) Budapesti Muszaki es Gazdasagtudomanyi Egyetem 4. (Fraunhofer) Fraunhofer-Gesellschaft zur Foerdering der Angewandten Forschung E.V 5. (UPM) Universidad Politecnica de Madrid 6. (VITO) Vlaamse Instelling voor Technologisch Onderzoek N.V. 7. (IIT) Technion – Israel Institute of Technology 8. (UPRC) University of Piraeus Research Center 9. (HU) University of Haifa
Contact details	Prof. dr. Davy Janssens *Universiteit Hasselt– Transportation Research Institute (IMOB)* **Function in DATA SIM:** Person in charge of scientific and technical/technological aspects **Address:** Wetenschapspark 5 bus 6 **|** 3590 Diepenbeek **|** Belgium **Tel.:** +32 (0)11 26 91 28 **Fax:** +32 (0)11 26 91 99 **E-mail:** davy.janssens@uhasselt.be **URL:** www.imob.uhasselt.be
Deliverable details	**Work Package** 1 **Deliverable:** D1.2 **Dissemination level:** xxx **Nature:** xxx **Contractual Date of Delivery:** 31.08.2013 **Actual Date of Delivery:** 31.08.2013 **Total number of pages:** 71 **Authors:** Nikos Pelekis, Panagiotis Tampakis, Marios Vodas, Stelios Sideridis, Despina Kopanaki, Daniel Schulz, Christine Kopp, Chiara Renso and Yannis Theodoridis
Abstract	This document is the second deliverable of WP1 (D1.2) that presents the results of the DATASIM project during Y2.

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

Content

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

1. Strategic Objectives of WP1

According to DATASIM's DOW document the strategic objectives of WP1 are:

WP1 aims to provide the big data infrastructure for storing, indexing, accessing, anonymizing, querying and analyzing the massive amounts of data needed by the project. This environment will feed the subsequent WPs with appropriate information for their purposes. Among the tools to be developed in WP1 are:

- *a novel extension of Hermes, called HERMES_sem, for the efficient management (storage, indexing, retrieval) of semantically-enriched mobility data; also with its privacy-aware 'skin', called HERMES_sem++, for handling data privacy aspects in both storage and retrieval phases;*
- *a crawler and an efficient data repository for social network (Facebook) data;*
- *a insightful network-based repository for power / energy consumption data;*
- *an integrated common environment – testbed – over big data, appropriately linking each other based on their context.*

(source: DATASIM technical annex, p. 12/109)

This document presents the results of the second (and last) deliverable of WP1 that have been prescribed to demonstrate its achievements. In detail, according to the designed workplan, the deliverable D1.1 has already been completed during the first year of the project. For details, please refer to D.1.1. During the second year, the WP1's research agenda was mainly driven by the development of an efficient and effective data manager and data warehouse infrastructure (i.e. D1.2). Again, as mentioned in the DOW document, D1.2 is about:

> *"D1.2) Final data manager and data warehouse storage infrastructure: A comprehensive data warehouse infrastructure is to be built in order to efficiently support storage and management of different big data types. The final version of the data manager and the data warehouse storage infrastructure is reported in this deliverable. [month 24]"*

WP1 has also been organized to three separate tasks. The first one (i.e. *Task 1.1: "Big Data acquisition, inventory and study area selection for scenario evaluation"*) has been already completed and its results are reflected in the D1.1 deliverable. As such, the second (i.e. *Task 1.2: "Big Data Repository"*) and the third (i.e. *Task 1.3: "Big Data Harmonization and Knowledge Integration"*) tasks correspond to the current deliverable (i.e. D1.2).

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

2. Executive summary of results/activities

In order to meet the above-described objectives, we have followed three lines of research which, although seem quite different, include the appropriate synergies to fulfill the envisioned goal:

- *Novel extensions of MOD engines to meet transportations and mobility mining fields*
- *Semantic enhancement of the data manager and the data warehouse infrastructure*
- *Meeting BIG data challenges for mobility data*

For the first two lines of research, the 'big picture' of the developed architecture of the data manager and data warehouse infrastructure is depicted in Figure 1. For the third line of research, we have designed a distributed architecture that uses as nodes the illustrated infrastructure, enhanced by novel segmentation methods that exploit off-the-self parallel processing techniques. We report on WP1's results/activities based on this figure.

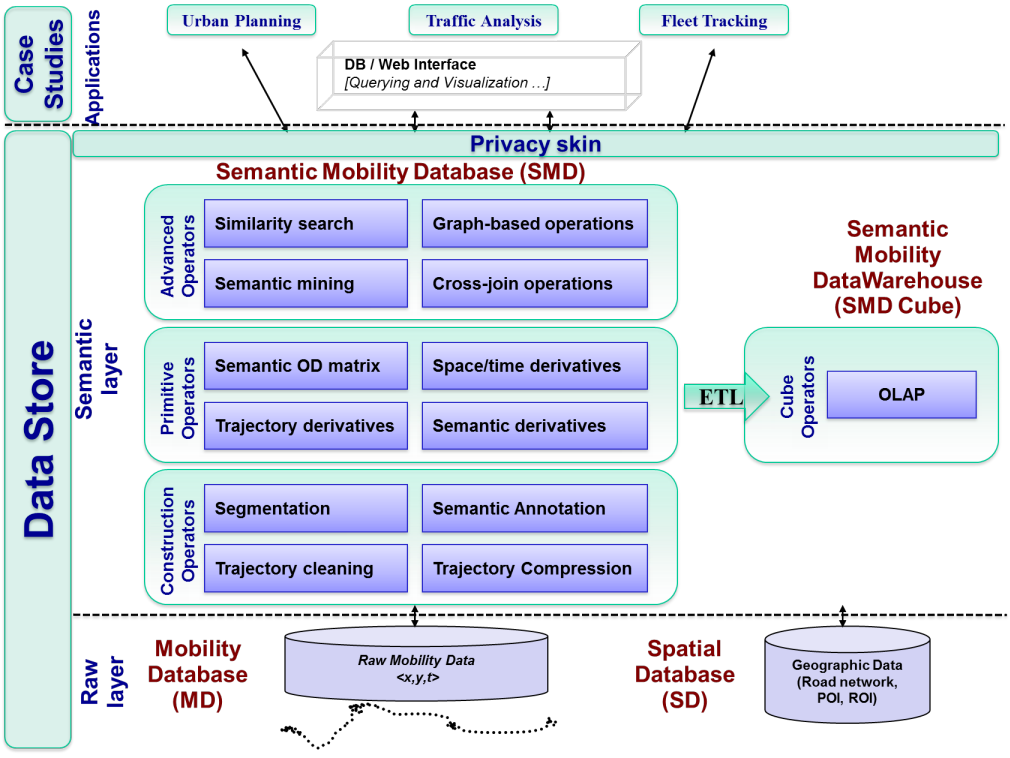


Figure 1: BIG data repository architecture

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

Novel extensions of MOD engines to meet transportations and mobility mining fields

Most of the envisioned applications, case studies in DATASIM require the historical information of the past movement (the trajectory) of the users, in various scales, levels of precision, formats, also accompanied with either non-spatial or other types of data. This implies that a so-called Moving Object Database (MOD) is required to support the various data management requirements. However, although the field of MOD has demonstrated its effectiveness in various scenarios, the challenges raised in DATASIM also called for novel interesting extensions. For instance, new functionality was required in terms of novel query operators, as well as new access methods to support the heterogeneity of the accompanied data and powerful query languages. The last decade, research in MOD focused only on the previous mentioned lines, at the same time that the transportation field started to use automatically collected mobility data, while the mobility data mining field was proposing effective analytical techniques for the discovery of various types of patterns. However these lines of research have not met so far. Given our earlier experience in the field, ***we designed and developed from scratch a highly efficient new MOD engine that is tailored to DATASIM purposes***. The result of this effort is demonstrated in Chap. 13 in [9], where the reader can see many typical examples of managing and querying a real trajectory dataset. This activity was the first step to meet this strategic objective.

The evolution of this effort was to ***bridge data management with data mining techniques***, focusing on the efficiency via the extension of MOD engines with mining functionality as query operators. This is an almost unexplored field in the data management and mining literature, which challenged us in DATASIM. To meet this requirement we focused on supporting clustering methods as build-in query operators inside the MOD engine [14][15]. The choice was not only that clustering is a prevailing method that can be used for various analysis tasks, as in our case it also plays the role of a smart partitioning method which is exploited to meet the big data challenges (see below). Furthermore, in order to support transportation analyses, we invented novel query types, which have been also embedded into MOD engines in order to enhance efficiency. Briefly, we introduced the so-called *"strict path query"* [11] and the *"trajectory buffer query"* [14], which can be used as building blocks for higher level data management tasks, such as (the afore-mentioned) mining operations, and transportation-oriented analyses [7].

Finally, we introduced the problem of map-matching trajectory compression [4] having as goal to have in hand a map-matching method (necessary to perform transportation-specific tasks) that also compresses the transformed data, without changing significantly the map-matched data, so as not to affect transportation analyses.

The above correspond to the first tier in the architecture of the developed data manager (see raw layer in Figure 1).

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

Semantic enhancement of the data manager and the data warehouse infrastructure

Storing and maintaining the history of a moving population in such an enhanced MOD and/or a Trajectory Data Warehouse (TDW), for operational and analytical processing, respectively, quickly results in huge volumes of data that make processing expensive and demanding in resources. If downsizing sounds to be the answer in the above problem, then extracting and managing semantics from (raw) trajectory data is a promising channel to do this. A semantically-annotated trajectory, in short semantic trajectory [41], is an alternative representation of the motion path of a moving object; Figure 2 illustrates the two alternatives. It is not only a matter of size of the database; maintaining semantic trajectories is quite useful in terms of movement analysis (which will be the key issue in next-generation LBS): movement is considered as a sequence of episodes (stops/moves) – e.g., home, shopping, move with bus, in train – which results in detecting homogenous fractions of movement, thus aiding mobility understanding. This representation of mobility tries to provide answers to questions of type when? where? how? what? why?

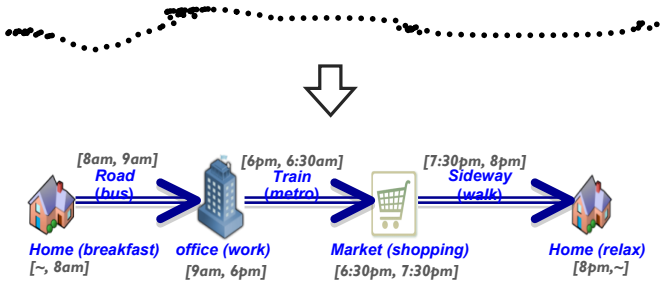


Figure 2: raw vs. semantic trajectories

On the other hand, what is missing is the DBMS support of this new type of information in order to realize the above model and give life to novel applications and services based on this. The most challenging aspect of this activity was that the data manager should handle complex objects, which are sequences of spatial-temporal + textual objects. This is exactly the goal of the research line that we followed: ***provide the infrastructure for semantically-aware MODs and TDWs, and their respective applications***. In detail, we designed and developed a type system with its associated query language to facilitate managing, querying, and analyzing such kind of data Chap. 9 in [9] and [12]. This effort resulted to to Hermes^{Sem} semantic-aware MOD engine, mentioned in the DOW annex. This engine has been successfully applied to real [13] as well as synthetic data Chap. 12 in [9]. The respective documents provide real hands-on experience upon the developed framework. What is more, we have designed a benchmark for such trajectory DBMSs [16], which will be applied to our Hermes system, also in comparison to an appropriately adapted PostGIS spatial DBMS.

In the same line, aggregated information from STDs stored in a Data Warehouse (DW), in the form of a graph-based representation of mobility-aware data cubes fed from a TD via an Extract-Transform-Load (ETL) procedure is another research line that we followed. The data cube paradigm has been

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

extended to support (raw) trajectory DWs [38], involving spatial and temporal versus thematic (alphanumerical) dimensions and spatial or spatio-temporal vs. numerical measures. Designing and building a corresponding semantic trajectory data cube following a graph-oriented representation was the outcome of this research activity [12].

Furthermore, the above described data infrastructure has been covered by a privacy skin [17] (the so-called Hermes^{Sem++}, as mentioned in the DOW annex) to support privacy-aware querying according to the lines prescribed in D1.1.

In order to experiment with the above framework also with synthetic data, beside the available real datasets, we developed a synthetic data generator, called Hermoupolis [8], which produces such sequences of spatial-temporal + textual information with user-defined properties. This activity further enables us to evaluate our work with data that correspond to objects moving with a transportation mode other than a car (note that such data are not available in DATASIM), while it allows us to produce synchronized (raw vs. semantic) data, which is important for validation of various DATASIM tasks (e.g. for validating the semantic enrichment techniques developed in WP2), as it contains ground truth data of the transformation implied in Figure 2. Finally, the generator allows us to produce user-defined volumes of data, which also facilitates efficiency and scalability experiments.

Another activity to meet this strategic objective was to explore narrowing the gap between the spatio-temporal aspects and the semantics involved that hinders trajectory analysis, benefiting from the growing collections of linked data, with well-defined and widely agreed semantics, already available on the Web. The result of this effort was Baquara [5], an ontology with rich constructs, associated with a system architecture and an approach to narrow this gap. The Baquara ontology functions as a conceptual framework for semantic enrichment of movement data with annotations based on linked data. The proposed architecture and approach reveal new possibilities for trajectory analysis, using database management systems and triple stores extended with spatial data and operators. The viability of the proposal and the expressiveness of the Baquara ontology and enabled queries are investigated in a case study using real sets of trajectories and linked data.

Finally, we proposed cost models for nearest neighbor query processing of existentially uncertain data [6], as semantically-enriched trajectories belong to this category of data.

Although semantic trajectory database management and analysis is still in its infancy, one can foresee the above research approach that will very soon foster new research efforts in the field. This line of research corresponds to the second (semantic) layer of Figure 1.

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

Meeting BIG data challenges for mobility data

It is evident from the discussion so far that the envisioned data manager implies a centralized architecture. In this line of research we investigated solutions to make the above-described infrastructure capable to deal with really BIG data, meaning that we explored for novel methods to exploit the power of the centralized data manager as part of a distributed environment. Towards this direction, we investigated several computing paradigms as distributed and parallel DBMSs or MapReduce-based solutions. To the best of our knowledge such an approach has only very recently started to be explored in the literature of mobility data. During the first year of the project we focused on technical preparations of this step, while during the second year we first studied exhaustively the corresponding literature Chap. 10 in [9] and then we devised appropriate segmentation / partitioning methods [14][15] that can be exploited by off-the-self distributed solutions. The idea is to ***be able to partition / distribute mobility data in a way that will take advantage of the local efficiency of the computation nodes***, by using the latter in off-the-self distributed and parallel solutions.

The above-described results correspond to activities that meet primarily the goals of Task 1.2 and secondary (some of them) the goals of Task 1.3. These results will be described in detail in the following three sections (i.e. 3, 4 and 5, one section for each research line that we followed). In addition (in section 6), we have worked extensively on the data harmonization and knowledge integration task (i.e. Task 1.3). More specifically, we worked on the setup of a spatial disaggregation process in the administrative district of Cologne, Germany. We extracted relevant data characteristics and identified linking variables and we performed the required clustering and transformation of the data. In addition, we prepared the framework for the execution of the statistical matching.

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

3. Novel extensions of MOD engines to meet transportations and mobility mining fields

The HERMES MOD engine

As it has already been discussed, in order to support the various data management requirements a MOD engine is required, which however has to be extended with novel and efficient functionality, so as to cover the challenges raised from the various heterogeneous datasets, as well as from the diverse specifications originating from the interdisciplinary nature of DATASIM. The starting point for this development was the state-of-the-art HERMES [42] MOD engine. Although the original HERMES has been proposed as a spatio-temporal extension of the object-relational DBMS of Oracle (also integrated with Oracle's Spatial component), in DATASIM we have redesigned and redeveloped the whole framework as extension of PostgreSQL with no other dependencies, (e.g. PostGIS spatial extension). This choice was made for two reasons. First of all, we wanted to move to an open-source framework, while additionally, the extensibility interface of PostgreSQL was more flexible and allowed us to investigate of successful indexing paradigms proposed in the literature, which were never applied and integrated into a real DBMS due to that is was either too difficult or impossible given the supported extensibility interface of the DBMS. So, from an abstract point of view, HERMES is a novel mobility-related extension of PostgreSQL that uses all of its extensibility interfaces, as depicted in Figure 3.

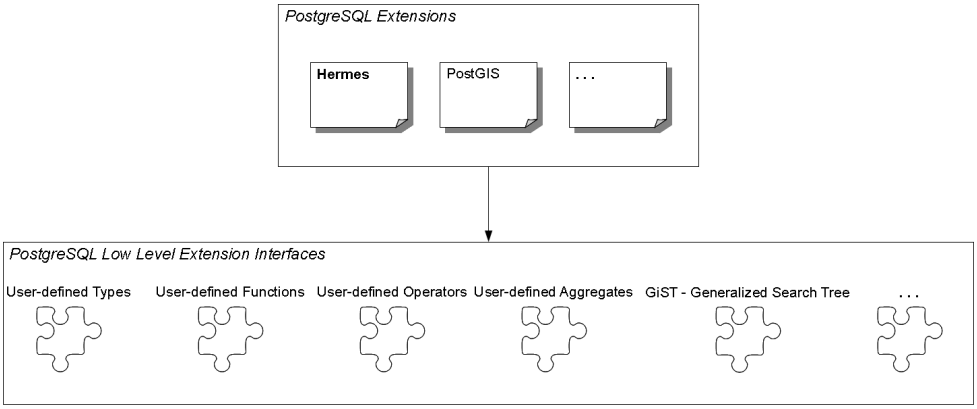


Figure 3: HERMES in PostgreSQL

Briefly, HERMES unifies spatial (OGC-compatible) and temporal dimensions, so as to deal with mobility data as first-class citizens. Overall, it is an SQL framework (collection of data types, functions, operators etc), where the end-user is required to know only SQL to use it. Since PostgreSQL provides mechanisms to write functions and data types in C or SQL, HERMES utilizes both ways in order to

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

take advantage of the efficiency of C and the simplicity of SQL. A representative sample of the mobility-related datatypes we have devised are illustrated in Figure 4.

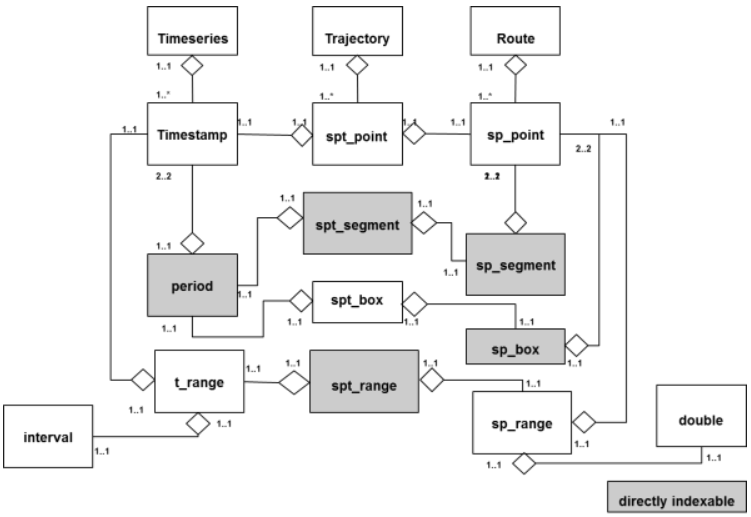


Figure 4: HERMES class diagram

Moreover, a family of appropriate custom access methods (AM) have been designed for these mobility datatypes. This family, called pg*X*D-Rtrees, where *X* in {1,2,3}, to support various dimensions, is based on the well-known Rtree extension, tailored for mobility data, however designed as variants of the Generalized Search Tree (GiST) AM. The reason for that is also the answer to the question, why high-quality mobility AMs well known in the literature, they have not been implemented in popular ORDBMS? The answer is due to the fundamental complexity and cost involved in developing AM and integrating them into database servers, especially their concurrency control and recovery components. GiST allows the development of such custom data types with the appropriate AM, by an expert in the domain of the data type, rather than a database expert. This is exemplified in Figure 5.

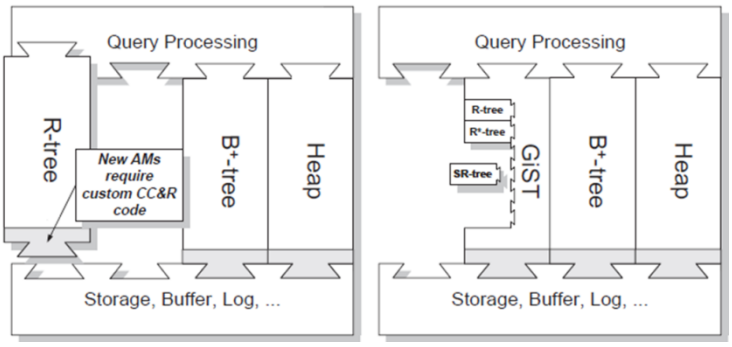


Figure 5: Extending GiST for mobility data

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

In order to support the various applications and case studies that will take place in DATASIM, but also aiming to the broad acceptance of the above framework by the research community, we have given it several degrees of freedom, which are illustrated in the architecture of Figure 6.

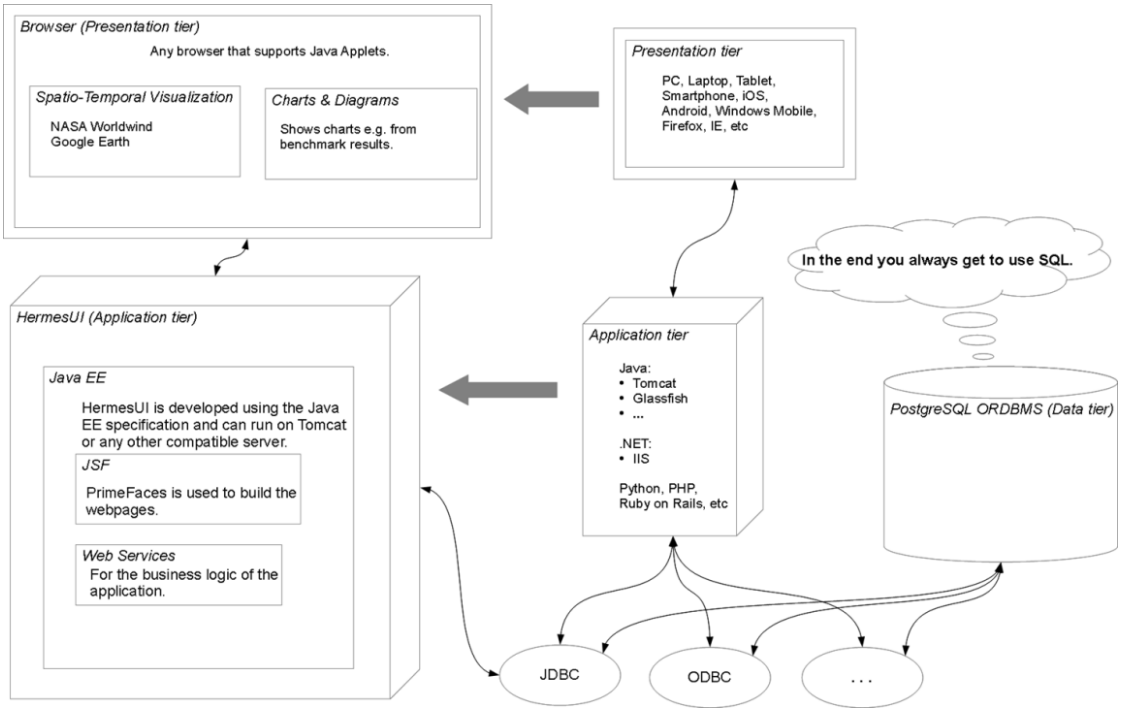


Figure 6: HERMES … from the user perspective

The functionality that is supported via HERMES's SQL API is quite rich. Of course, the palette of functions and operators includes traditional types of MOD queries such as:

- Trajectory reconstruction/cleaning methods
- Methods per Individual/Trajectory
  - spatial & spatio-temporal projection operation (e.g. at instant/period/area)
  - space/time derivatives (e.g.Enter/Leave points/timestamps w.r.t. an area)
  - statistical functions (e.g. avg/min/max velocity/acceleration, radius of gyration etc)
  - spatio-temporal compression techniques(e.g. Douglas-Peucker and TD-TR algorithms)
  - a library with most of the well-known distance (similarity) functions
- Index-supported queries
  - Timeslice
  - Range

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

- NN vs. static reference objects

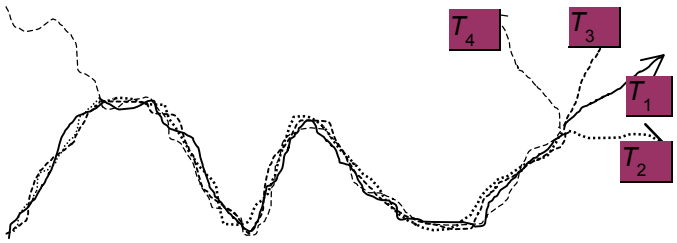
An interesting extension of HERMES is that the above implied query language is applicable for objects moving in an unconstrained space, as well as for network-constrained moving objects, as such introducing for the first time a MOD engine for such moving objects. This particular functionality has been incorporated via the use and extension of the pgRouting module.

More interestingly, HERMES includes a wide list of operations that extends its query language with semantics inspired from the transportation and mobility mining fields. Incorporating the subsequent list of operations does not only extend the query language with well-known or new DATASIM-required functionality, but the focus here is on the invention of efficient methods to do so.

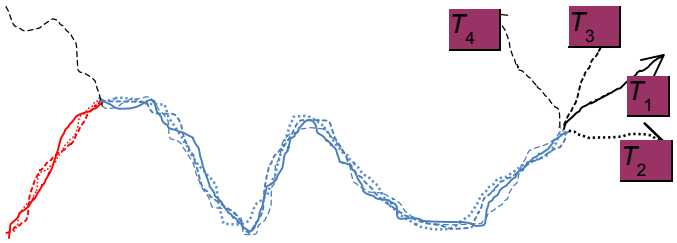
- ***Potential Area of Activity - PAA*** [44]: With this query the user may identify the region (and as such the Points of Interest – PoI inside it) where the user could have been between a known starting and ending position and some assumptions, such its maximum speed. The PAA, defined by time geographers, is very useful in the case of very low sampled or highly uncertain data, such as the diaries that we have in our possession.
- ***Emissions and energy consumption from GPS data*** [40]: With this query we may associate GPS data with the corresponding emissions and energy consumption and thus we may have point to point, map-matched quantification of these measures.
- ***Origin-Destination (OD) or path matrix***: With this operator we deal with the problem of the efficient population of an Origin-Destination (or path) matrix, which is one of the most useful tools in transportation science. More specifically, we try to tackle the problem in its various forms. For example, the region of interest could be divided in a) Discrete Areas b) Overlapped Areas, or c) Fully Partitioned Areas (e.g. *N*x*N* boxes). Each element (*i*,*j*) of the Origin-Destination Matrix could contain: i) which (or how many) objects started from a specific area *i* and ended up at another specific area *j*, ii) which (or how many) objects overlapped a specific area *i* and subsequently overlapped another specific area *j* during their course. Obviously, one way to resolve this query is to devise an algorithm whose kernel is a bunch of traditional spatio-temporal queries.
- ***Whole vs. Sub- Trajectory clustering***, namely *T-OPTICS* [37] and *Sampling-based Sub-Trajectory Clustering* [14], respectively: These operators tackle efficiently the problem of whole and sub-trajectory clustering (and outlier detection), but as query operators in a real MOD. For whole trajectory clustering, we chose the state-of-the-art T-OPTICS algorithm, and we extended it by integrating it with the library of different distance functions mentioned above. Regarding the sub-trajectory problem, we devised a novel solution based on the concept of representativeness in MOD [39] by developing an algorithm that operates in 4 phases, which in turn imply 4 different activities on which we are working: a) voting methods to define representativeness of a trajectory w.r.t. other trajectories, b) trajectory segmentation based on representativeness, c) sub-trajectory sampling to operate only on a small MOD, sub-trajectory clustering based on segmented and sampled sub-trajectories.

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

Figure 7 illustrates an example of the clustering of our methodology applied to a MOD comprised by four trajectories, T_1, …, T_4. (In this figure, the time dimension has been ignored for visualization reasons.) Our proposal is able to identify two clusters of sub-trajectories (in red and blue) and five outlier sub-trajectories (in black). On the other hand, the current state-of-the-art technique for sub-trajectory clustering, TRACLUS [35] would first simplify the trajectories into large line segments and then apply a grouping of similar line segments, thus, as delineated by the authors, discovering linear patterns only and failing to identify complex (e.g. snake-like) patterns like the ones that appear in Figure 7. In other words, when applied to this toy MOD, TRACLUS would eventually result in the discovery of six linear clusters (one cluster for each time the snake-like motion changes direction). On the contrary, according to our approach, the clustering uses the results of segmentation and sampling processes that take into account the spatiotemporal similarity. In addition, there is no constraint on the complexity of the shape of sub-trajectories found, yielding a clustering that is related only on line segment representativeness. Furthermore, we should note that we do not pose any geometrical constrains in terms of parameters of the proposed methodology as those required by moving clusters, the various types of flock patterns. This type of parameters, such as the radius of disc, or the (max) duration and the cardinality of the patterns are hard to be defined, while their tuning may result in completely diverging outcomes.



(a)



(b)

Figure 7: (a) a MOD of four trajectories; (b) two identified clusters of sub-trajectories (in red and blue) and five outlier sub-trajectories (in black).

The design of such a clustering methodology that will be free of shortcomings like the above, is subject to two more requirements which challenged our research: (a) we seek for an efficient and scalable solution, which (b) should be based not on an ad hoc implementation based on a sophisticated access

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

method, but rather on a real-world MOD management system so as to be useful in real-world application scenarios, where concurrency and recovery issues should be taken into consideration. As such, the whole process is boosted by the indexing mechanisms previously mentioned. At this point, we hide the details of the clustering method, which will be given subsequently (in Section 5), when presenting our approach for tackling the BIG data challenges, as this clustering method also acts as a partitioning method, that is a mandatory step to move from a single-server to a distributed architecture.

The preliminary results for the above described MOD framework are very encouraging, not only in terms of functionality and usability, but also in terms of efficiency, showing a significant speedup of crucial operations (such as range queries, which are the core for many other operations, e.g. OD matrix and clustering algorithms computation) w.r.t. PostGIS. Such kinds of experiments are only examples from a complete benchmark [16] that we have designed for WP4 purposes but also for evaluation purposes of WP1. We leave the details of this activity to be presented at the corresponding deliverable of WP4. In Chap. 13 in [9] we provide a hands-on experience of Hermes that includes many examples demonstrating the application of the afore-mentioned functionality to a real trajectory dataset.

Interactive Traffic Analysis with the Strict Path Query

As mentioned earlier, in order to support transportation analyses, we invented a novel query type, called the “strict path query” [11], which allows us to perform transportation-oriented analyses [7] over large datasets, extremely efficiently. Trajectories of moving objects in a road network are excellent data sources for traffic analysis, because the path of trajectories can be included into the analysis. Due to this, it is possible to study multi-edge effects such as turn-time costs and signal coordination. Several measures are proposed in [54], for path-based traffic analysis (travel time, congestion index, proportion stopped time, and acceleration noise) that provide unique insights into the traffic conditions. In order to evaluate any such traffic measure it is necessary to retrieve the trajectories that follow a specific path in the road network. State-of-the-art in network constrained indexing [46] supports path queries, but cannot guarantee that the retrieved trajectories follow the entire path without detours. Filtering out trajectories with detours is important for many use cases, such as computing travel-time, fuel-consumption, or analysis of signal coordination. However, retrieving the trajectories that follow a specific path through the road network is inefficient with state-of-the-art, and as a consequence, so is evaluating many path-based traffic parameters. Consider computing the average travel-time of the thick dashed path in Figure 8 (the main road Vesterbro in Aalborg, Denmark). Issuing path query for the trajectories that touch each edge in the path results in 2184 trajectories from our data set. Of the retrieved trajectories only 1804 actually follow the dashed path without detours. The remaining 380 trajectories follow one of the thick solid paths. Obviously, using all 2184 trajectories directly in any analysis of the thick dashed path will compromise the validity of the analysis.

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

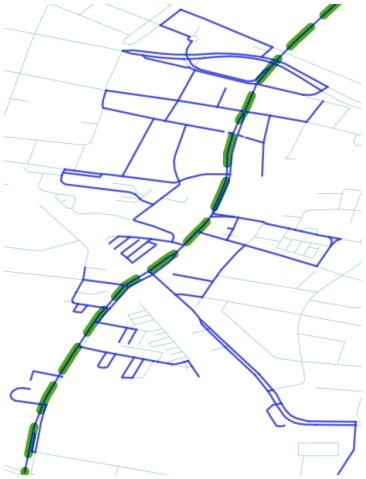


Figure 8: The trajectories retrieved for the thick dashed path.

Intuitively, a path-based travel time estimate can be found by computing the average travel time of each edge in the path and sum up these travel times. However, such an edge-based estimate does not capture important effects such as signal coordination and turn-times. To illustrate this, we have retrieved all trajectories that follow the paths *abc* and *abd*, respectively, in Figure 9. The average travel times of the edges *a* and *b* is computed using the respective trajectory sets (see the table in Figure 9). As can be observed, the average travel times of these edges strongly depend on the direction through the intersection. Therefore, including a trajectory such as the one shown by red circles will not yield accurate travel time estimates for the path *abd*. The example in Figure 9 is one out of numerous cases in our real-world data set that we used. Several other measures, e.g., fuel consumption, acceleration noise, and congestion index are correlated with travel time, and thus likely to be affected similarly.

path *abc*		
edge	time	count
a	48	829
b	48	829

path *abd*		
edge	time	count
a	36	587
b	41	587

Figure 9: Edge based average travel times for trajectories following abc or or abd, in a small snippet of the road network in Aalborg, Denmark. a, b, c, and d are edge identifiers.

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

To distinguish between path queries in the related work and the variation targeted in this work, we named our variation the Strict Path Query (SPQ). A SPQ can be evaluated using state-of-the-art by retrieving the trajectories that touch each edge in the path. The intersection of these result sets form a candidate result set, i.e., the trajectories that touch all edges in the path queried. The result set may include trajectories with detours, which can be removed by retrieving the full trajectory data, or by verifying that the edges adjacent to the SPQ are not visited by any of the candidate trajectories. This approach is inefficient in that for every edge in the path it is necessary to retrieve all trajectories touching that edge.

In this work, we proposed the Interactive Traffic Analysis (INTRA) methodology for interactive path-based traffic analysis. The basic idea behind INTRA, is to represent each trajectory as a sequence of touched network edges. For each edge touched by a trajectory, an entry is generated containing the id of the edge, enter and leave time, and a novel encoding of the entire path leading up to and including that edge. The benefit of the path encoding is that given two entries from a trajectory, it is possible to determine whether the trajectory followed a particular path between the two entries without retrieving the complete intermediate trajectory data. As such, aggregate measures for paths, e.g., travel time and fuel consumption can be evaluated by retrieving the trajectory data for only a few edges in the path.

INTRA is implemented in standard SQL and therefore highly portable between DBMSs. In addition, INTRA is easy to integrate with state-of-the-art in network constrained indexing [46]. Using our methodology we demonstrated speed-ups over state-of-the-art of up to two orders of magnitude for SPQs. We performed extensive experiments using a real-world trajectory data set containing 200 million GPS records and the entire road network for Denmark. The proposed methodology does not require any expensive pre-processing of the road network, contrary to state-of-the-art. More details of this activity can be found in [11][7].

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

4. Semantic enhancement of the data manager and the data warehouse infrastructure

Coming back to the discussion about the management and analysis of semantic trajectories and the lack of DBMS support for them, it is clear that raw and semantic trajectories do not have much in common. Technically speaking, a (raw) *trajectory* is a model for a motion path of a moving object (animal, car, human, etc.). Due to discretization, it is usually defined as a sequence of sampled time-stamped locations (p_i, t_i) where p_i is a 2D point (x_i, y_i) and t_i is the recording timestamp of p_i. On the other hand, a semantically annotated trajectory, in short *semantic trajectory* [41], is an alternative representation of the motion path of a moving object; Figure 2 illustrates the two alternatives: a raw trajectory consisting of original GPS records (top) and its semantic counterpart that contains meaningful and easy to be understood abstractions of mobility data (bottom).

Maintaining raw trajectory information in a Moving Object Database (MOD) quickly results in huge volumes of data that make processing an expensive and demanding task. For example, only keeping track of a (modest) population of 10,000 vehicles driving on the average for 2 hours daily with a sampling rate of 1 record every 5 secs, results in a database of more than five billion records (order of several Tbytes) after one year of tracking. If downsizing sounds to be the answer in the above problem, then extracting and managing semantics from (raw) trajectory data is a promising channel to do this.

It is not only a matter of the database size; maintaining semantic trajectory information turns out to be quite useful in terms of content-related movement analyses (which is one of the key issues in DATASIM). In the widely adopted semantic trajectory terminology [41], movement is considered as a sequence of semantically meaningful episodes, either stops (e.g. at home, at office, for shopping) or moves (walking, driving, etc.), which results in detecting homogenous fractions of movement. Such semantic-aware mobility fractions and abstractions enable applications to better understand and exploit on human mobility: for instance, identify those meaningful locations where some activity (work, leisure, relax, etc.) takes place, infer how long does it take to get from one *place of interest* (POI) to another (e.g. from home to office) using which transportation means, conclude about the frequency of an individual's outdoor activities (recorded e.g. by a fitness application), calculate indices related to environmentally friendly or sustainable mobility. In addition, applications built upon semantic trajectories can easily identify common behaviors among a group of people. As such, future semantic-aware LBSN applications can be established, including location sharing and ranking, recommendation according to travel and socio-demographic similarity – e.g. car-pooling applications – etc.

Although conceptually strong, such a representation of mobility lacks a robust DBMS support in order to realize the above model and give life to novel applications and services based on this. This implies a unified framework that will enable efficient and effective storage, indexing mechanisms of such heterogeneous information, as well as extended query languages to support novel types of queries and advanced methods for mining and multidimensional analysis. The following section argues for such a framework, while subsequently we present its core constituents, namely *lifesteps* and the

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

resulting *semantic mobility timelines*, the representation of the latter in *semantic mobility networks* and their further aggregations in *semantic mobility cubes*.

Motivation for a semantic-aware mobility data management framework

Let us consider the following scenario: humans living in a town are moving, stopping either by purpose (for doing an activity) or accidentally (e.g. due to a traffic light), and their movement is recorded by GPS-enabled devices and/or delivered by people themselves through diaries (usually collected for transportation research purposes). On the one hand, we have collections of GPS data that represent mobility in terms of spatial coordinates while, on the other hand, we have mobility information in terms of responses to where? when? what? why? etc. questions; see e.g. an example of a diary in Figure 10.

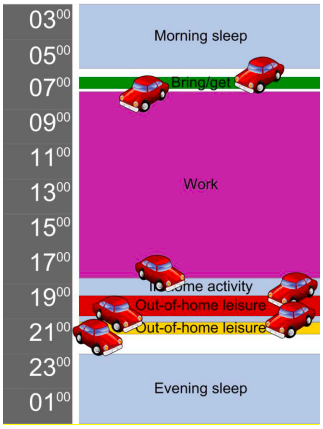


Figure 10: A diary [33]

These data collections, either independently or in conjunction, are useful for "what-if" analysis. For instance, queries like "*Find people who follow the pattern home–office–home from Monday to Friday*" or "*Find people who cross the city center on their way from office back to home*" or "*Find how many people make long trips (e.g. more than 20 km) on their way from home to office without taking into consideration the trajectories which include intermediate stops*" or "*how much time do parents spent for bring-get activities at the schools of their children?*" are obviously extremely useful in the transportation and urban planning domains and of course in DATASIM objectives.

Modeling, management and knowledge discovery aspects on (raw) trajectory data have been exhaustively researched in the past two decades [29], including plenty of algorithms and systems, spread from data management ([55][42][30]) to data mining [19][26].

On the other hand, semantic mobility data management is a relatively new entry in the research agenda. Models for semantic trajectories include [50][20], while techniques for extracting semantic trajectories from raw ones have been also proposed recently [59][62][58][60]. In [41] the interested reader may find a complete survey of relevant models and techniques.

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

Generally speaking, a semantic trajectory is defined as a sequence of *episodes*, labeled either as *Stops* or *Moves*, each with associated meta-data (*tags*). Technically, *Stops* are places (points, regions) where the object remains "static", *Moves* are the parts of the object's trajectory in between two Stops, i.e., where the object is "moving", and *tags* are meta-data associated with *Stops* and *Moves*. According to the state-of-the-art model [41], a semantic trajectory always starts and ends with a *Stop*, while *Moves* are found in-between two *Stops*. This modeling approach is quite restrictive, as it cannot effectively represent trips in diaries used in the transportation domain. In this domain, diaries only imply trips that are executed for a particular purpose. Here, travel and transport should be seen as a "derived" demand from the demand of activities. In this definition and the ruling utility-maximization literature which is underlying, people perceive transport as a negative utility that needs to be done in an attempt to satisfy needs and purposes in life which are associated with a positive utility. As such, diaries do not contain "accidental" stops like for instance stops due to traffic congestion. However complex patterns like successive *Moves* (e.g. due to multimodal transport like for instance bus-train) may also appear.

Apart from relaxing the above limitation, a generic model for semantic trajectory modeling should also be able to support modeling at various scales and/or spatio-temporal granularities. Think, for example, of a person being at a shopping mall: this activity can be modeled as a *Stop* but novel applications would benefit if the model allowed drill-down (zoom-in) inside this *Stop* and examine the micro-mobility pattern of the individual (i.e. the shops that she visited during her stay there). The challenge adopted in this work is to address the above requirements at the conceptual level by introducing the concepts of *Mobility Timelines* and *LifeSteps*, and then provide the seeds for a so-called semantic-aware *Semantic Mobility Database* (SMD) and its aggregation in *Semantic Mobility Cubes* (SMC) (to be defined later).

Figure 1 sketches the big picture of our objective:

- at the bottom-layer, the previously presented MOD lies, being in charge of the raw mobility data and supported by the discussed access methods and query functionality;
- at the middle-layer, it is the SMD that provides novel datatypes, indexing methods, and operators extending MOD query languages for querying and analyzing mobility data from a semantic perspective.
- at the top layer, the application interface provides users with querying and analysis functionality on either MOD or SMD.

From Semantic Trajectories to Mobility Timelines

Hereafter, we present a novel conceptual model that resolves the shortcomings of the semantic trajectory model used so far in the literature. First, we provide a definition for raw trajectory (and sub-trajectory), upon which we define the core concept of our model: the so-called *semantic mobility timeline*. Figure 11 illustrates two typical semantic mobility timelines (with blue and red color, respectively) that will serve as the running example for the discussion that will follow.

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

Definition 1 (raw trajectory): *a (raw) trajectory τ of a moving object is defined as a triple (o-id, traj-id, T), where o-id (traj-id) is the identifier of the moving object (the specific trajectory of the moving object, resp.) and T is a 3D polyline consisting of a sequence of N+1 pairs (p_i, t_i), $0 \leq i \leq N$, assuming linear interpolation between two consecutive pairs (p_i, t_i) and (p_{i+1}, t_{i+1}) where p_i is a 2D point (x_i, y_i) in the plane and t_i is a timestamp.*

Definition 2 (raw sub-trajectory): *a (raw) sub-trajectory τ' of a (raw) trajectory τ is defined as a quadruplet (o-id, traj-id, subtraj-id, T'), where o-id (traj-id, subtraj-id) is the identifier of the moving object (the specific trajectory and sub-trajectory of the moving object, resp.) and T' is the portion of T between two timestamps, t_i and t_j, $t_i < t_j$.*

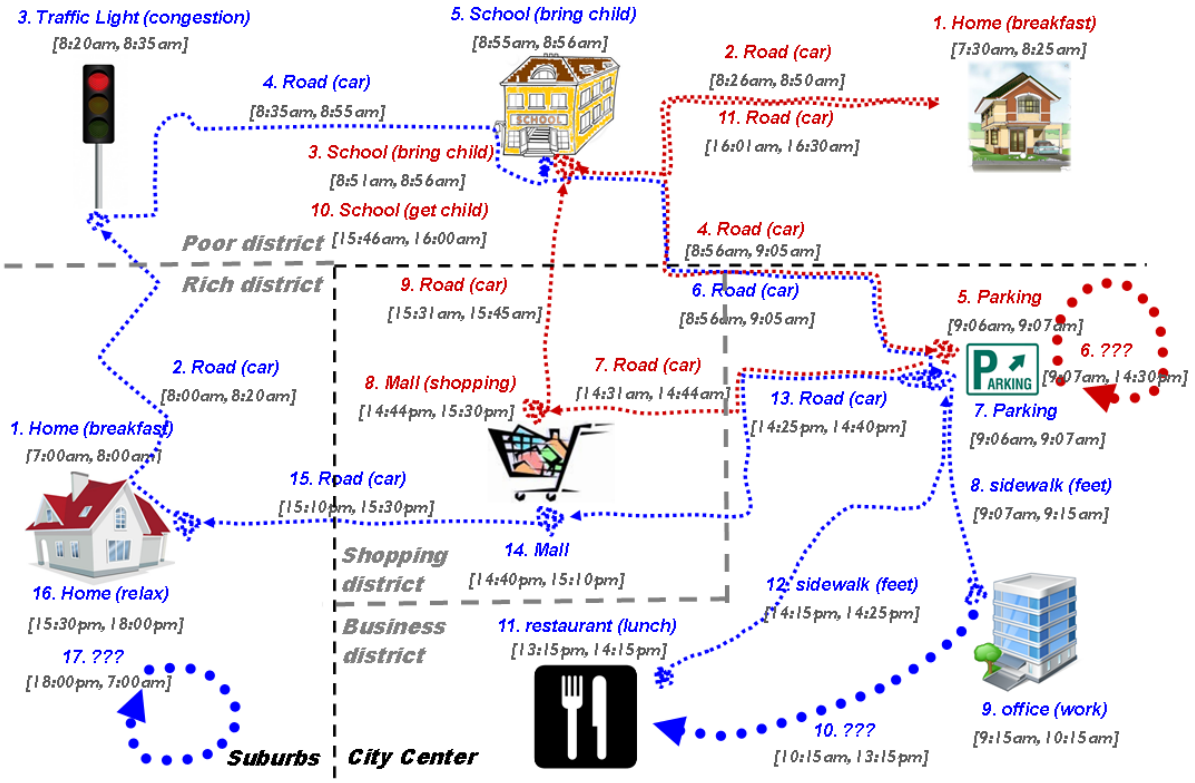


Figure 11: the semantic mobility timelines of two moving objects during a day

Having the above definitions in our hands, we define the necessary concepts towards semantic variants of the latter.

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

Definition 3 (Stop): *a Stop corresponds to a sub-trajectory τ' with specific spatio-temporal properties. In particular, a sub-trajectory τ' is labeled as 'Stop' iff its spatial (temporal) projection obeys a predefined spatial (temporal, resp.) constraint C_{space} (C_{time}, resp.).*

Obviously, the constraints responsible for characterizing a part of the trajectory of a moving object as *Stop* (C_{space} and C_{time}, resp.) are related to the spatiotemporal properties of the object and are application-oriented. Examples of C_{space} could be: "… lies within a circle of radius less than a threshold σ"; "… lies within distance less than a threshold σ from a POI"; etc. On the other hand, an example of C_{time} could be "… is an interval of duration more than a threshold δ".

Definition 4 (ActivityStop): *an ActivityStop corresponds to a sub-trajectory τ' with specific spatio-temporal and thematic properties. In particular, a sub-trajectory τ' is labeled as 'ActivityStop' iff its spatial (temporal) projection obeys a predefined spatial (temporal, resp.) constraint C_{space} (C_{time}, resp.) and it obeys a predefined thematic constraint $C_{thematic}$.*

Conceptually, an *ActivityStop* differs from a *Stop* in that the spatio-temporal constraints (i) are not necessarily identical and (ii) should be accompanied (in the case of *ActivityStop*) by an activity which takes place at the region defined by the spatial constraints, implying the purpose for being located at that specific place at that specific time (in different words, *the thematic properties imply the purpose for which the object moved to the corresponding spatial area*). On the contrary, a *Stop* may correspond to a forced or unintentional stop of moving object's transposition. In the example of Figure 11, stopping at the parking area or at the traffic light correspond to a *Stop*, while stopping at e.g. a school for dropping a child is an *ActivityStop*.

Definition 5 (MeteorStep): *a MeteorStep corresponds to a sub-trajectory τ' whose T' component (i.e., the corresponding 3D polyline) is unknown or empty (aka NULL). Furthermore, a MeteorStep is labeled as:*

- *'StopGap', iff the last known location of the moving object before the sub-trajectory in question, and the first known after that, coincide w.r.t. a predefined spatial and a predefined temporal predicate (gap_{space} and gap_{time}, respectively);*
- *'MoveGap', otherwise.*

Examples of predicates mentioned in the above definition could be: "*the distance between the two sub-trajectories is less than 100m"* (gap_{space}) or *"the unknown movement lasts for more than 3 hours"* (gap_{time}). Intuitively, a *StopGap* corresponds to a period of time where there is lack of information for the raw trajectory of a moving object, however it can be argued that the object remains static by considering the places before and after the gap; consider, for example, a user who switches off her GPS when returning home in the evening and switches it on again the next morning, see blue trajectory in Figure 11. (Obviously, in this case we cannot infer whether the object has moved during the in-between period of time.) On the other hand, in a *MoveGap* we are confident that the object has moved (comparing the before and after statuses) but we cannot argue how or when or by following which route it moved; see e.g. the transposition from office to restaurant in Figure 11.

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

Definition 6 (Move): *a sub-trajectory τ' is labeled as 'Move' iff it is neither a Stop nor an ActivityStop nor a MeteorStep.*

At this point, we have been able to label a portion of movement as either *Stop* (with the distinct case of *ActivityStop* included) or *Move*, also providing support to special cases that may happen due to lack of information (*MeteorStep*).

Definition 7 (LifeStep): *a LifeStep ls corresponds to a sub-trajectory τ' and is defined as a tuple (LifeStepID, LifeStepFlag, MBB, tags, T-link, Z-In, Z-Out), where:*

- *LifeStepID is the identifier of the LifeStep,*
- *LifeStepFlag is a flag taking values from the set {'Move' 'Stop', 'ActivityStop', 'StopGap', 'MoveGap'} (i.e., τ' falls into one of the corresponding definitions presented above),*
- *MBB is a tuple (MBR, t_{start}, t_{end}) corresponding to the 3D approximation of τ', with MBR being the 2D enclosing rectangle of the spatial projection of τ' in 2D plane and [t_{start}, t_{end}] being the 1D temporal projection of τ' in 1D timeline,*
- *tags is a set of keywords, describing the corresponding activities and semantic annotations related to this portion of movement,*
- *T-link is a link to τ',*
- *Z-In and Z-Out are pointers to semantic mobility timelines (defined below), which allow representation of a LifeStep at different granularities.*

Actually, *Z-In* and *Z-Out* attributes imply a recursive definition in order to support the zoom- in/out in different spatial scales (e.g. an *ActivityStop* at a shopping mall may be analyzed at a more specific level by looking at the shops where the user visited).

At this point, we are ready to provide the definition of semantically-annotated (in short, *semantic*) *mobility timeline*.

Definition 8 (semantic mobility timeline): *a semantic mobility timeline τ_{sem} of a moving object is defined as a triple (o-id, timeline-id, T_{LS}), where o-id (timeline-id) is the identifier of the moving object (the semantic mobility timeline of the moving object, resp.) and T_{LS} is a sequence of LifeSteps belonging to the same trajectory τ and being successive in time, i.e., $s_i[t_{end}] = s_{i+1}[t_{start}]$.*

Back to our running example of Figure 11 and according to the above definitions, e.g. the blue semantic mobility timeline consists of a sequence of 17 *LifeSteps* as follows (only *LifeStepIDs* and *LifeStepFlags* are listed):

<1. ActivityStop; 2. Move; 3. Stop; 4. Move; 5. ActivityStop; 6. Move;
7. Stop; 8. Move; 9. ActivityStop; 10. MoveGap; 11. ActivityStop; 12. Move;
13. Move; 14. ActivityStop; 15. Move; 16. ActivityStop; 17. StopGap>.

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

What we can see is that the proposed model is generic compared to state-of-the-art [41] since it relaxes some key limitations of the latter. According to [41], (i) a semantic trajectory (semantic mobility timeline, in our terminology) should start and end with a *Stop* (*Stop* or *ActivityStop*, in our model) episode and (ii) it is not allowed to have successive *Move* episodes. By relaxing these limitations and providing more alternatives than the typical *Stop* and *Move* episodes makes our model strong enough to support several application-oriented requirements.

By comparing Definitions 1 and 8, it is clear that the content of a MOD (raw trajectories) and that of a SMD (semantic mobility timelines) do not have much in common. This means that an existing MOD engine, such as the state-of-the-art and Hermes [42] and Secondo [30], cannot be used as-is in order to handle such SMD. Moreover, as we are interested in a unified framework wherein both raw and semantic data exist, efficiently and effectively querying MOD and SMD repositories is also challenging. Recalling the transportation example discussed in the previous section, three types of queries are foreseen:

- Q1 (queries on raw trajectories): queries involving MOD;
- Q2 (queries on semantic mobility timelines): queries involving SMD;
- Q3 (cross-over queries): queries involving both MOD and SMD.

As already discussed, Q1-type queries (range, nearest-neighbor, enter, cross, etc.) have been extensively studied in the MOD literature. On the other hand, Q2- and Q3-type queries are innovative and they cannot be considered as straightforward variations of Q1-type queries.

From Mobility Timelines to Semantic Mobility Networks and Cubes

Traditionally, aggregated information from DBs is stored in a Data Warehouse (DW), in the form of data cubes [28]. Data cubes are views of a DW, used for multi-dimensional analysis, the so-called On-Line Analytical Processing (OLAP). The data cube paradigm has been extended to support spatial [31], and (raw) trajectory DWs [38], involving spatial, temporal and thematic dimensions as well as spatial, spatio-temporal and numerical measures.

As a step forward, we introduce the notion of *Semantic-aware Mobility Data Cube* (SMDC), where aggregated data should not only expose interesting measures w.r.t. the chosen dimensions via a relational format, but they should directly encapsulate the spatial topology and its intrinsic relationships. To succeed this ambitious goal, we first define the so-called *Semantic Mobility Network* (SMN), a dynamic graph representation of the semantic mobility timelines stored in an SMD.

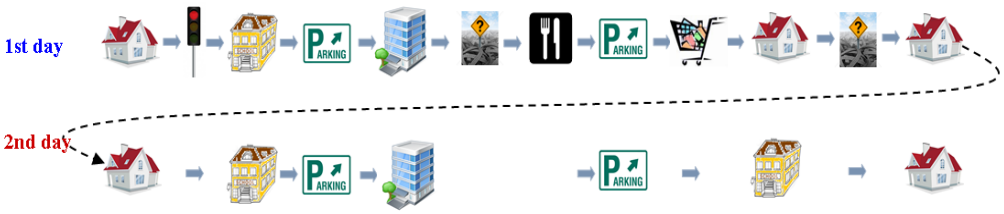
To illustrate the idea of SMN, Figure 12(a) presents the timeline of the blue trajectory of Figure 11 extended to include a 2-days period. Typically, during the second day, the user may follow a slightly different timeline. Figure 12(b) illustrates a graph representation of this timeline, in which vertices and edges correspond to non-*Move*- and *Move*- type *LifeSteps*, respectively. The “weights” of vertices and

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

edges are sequences of tuples describing interesting properties of the corresponding *LifeSteps*, such as distinct number of timelines or users, distinct number of activities, average speed, radius of gyration, potential area of activity, etc.) The cardinality of the sequence of tuples corresponds to a discretization of the time domain. In Figure 12(c), we depict an aggregate view of the SMN, where the aggregation occurs in the temporal dimension.

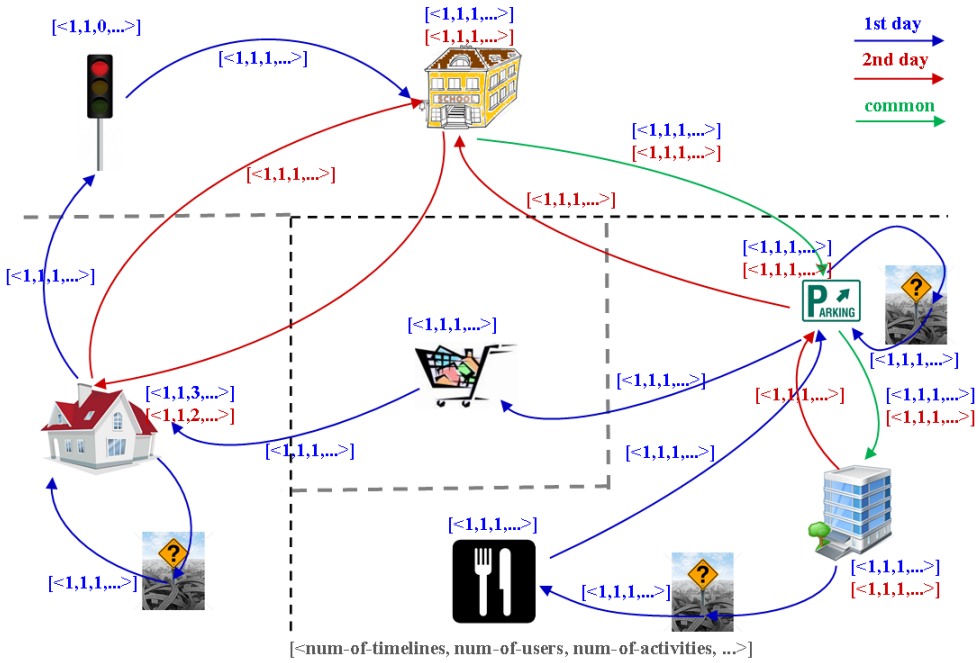
A nice interesting characteristic of a SMN, which we believe is a breakthrough for the mobility management and analysis domain, is that this graph-based design is data-driven and unifies all the mobility-related dimensions (space, time, semantics). Below, we provide formal definitions of a SMN and an aggregate SMN:

Definition 9 (semantic mobility network): *A semantic mobility network N, is a graph denoted by N = (V, E, M), where V is a set of vertices, $E \subseteq V \times V$ is a set of edges and $M = \{M_1, M_2, \ldots, M_n\}$ is a set of measures applicable to vertices and edges, i.e. $\forall v \in V$ and $\forall e \in E$, there is a tuple M(v) of v and M(e) of e respectively, denoted as $M(v) = (M_1(v), M_2(v), \ldots, M_n(v))$ and $M(e) = (M_1(e), M_2(e), \ldots, M_n(e))$, where $M_i(v)$ and $M_i(e)$ is the value of v and e on i-th measure, $1 \leq i \leq n$. The set V of vertices corresponds to the union of all distinct LifeSteps that are not of Move type (i.e., either Move or MoveGap), of all mobility timelines τ_{sem}, while the set E of edges corresponds to the union of the Move type LifeSteps. The set M of measures is a set of scalars quantifying properties of vertices and edges.*

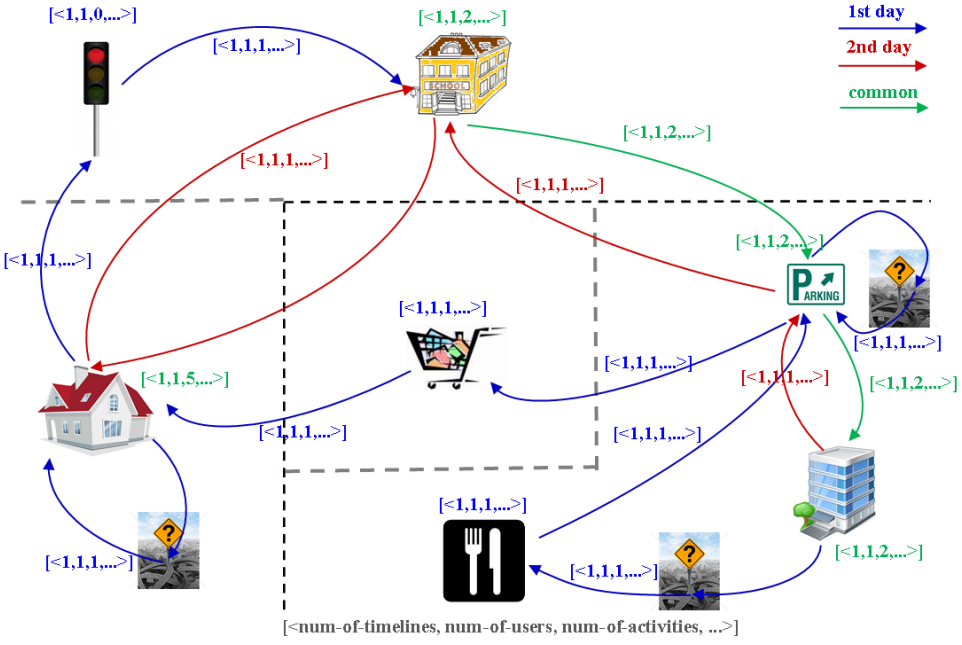


(a)

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833



(b)



(c)

Figure 12: (a) The abstract sequence of a 2-days semantic mobility timeline (b) its mobility network, and (c) its aggregate (over a period of time) network.

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

Definition 10 (aggregate semantic mobility network): *Given a semantic mobility network N, a set of dimensions D = {D_1, D_2, …, D_m} with their corresponding hierarchies, an aggregation D^a = {D^a_1, D^a_2, …, D^a_m} along these hierarchies, with $D^a(v)$ = ($D^a_1(v)$, $D^a_2(v)$, …, $D^a_m(v)$) denoting a tuple of values $D^a_j(v)$ of v on j-th dimension, $1 \le j \le m$ (similarly, $D^a(e)$ is defined), upon which measure M_i can be aggregated, an aggregate semantic mobility network w.r.t. D^a is a semantic mobility network N^a = (V^a, E^a, M^a), where:*

- *V^a is the set of aggregate vertices $v^a \in V^a$, each of which is constructed by a unification process $U_V([v])$ upon a nonempty equivalence class [v] of V, where $[v]=\{v|\ D^a_j(v)= D^a_j(u),\ v,u \in V,\ j=1..m\}$,*
- *E^a is the set of aggregate edges $e^a \in E^a$, each of which is constructed by a unification process $U_E(E(v^a, u^a))$ upon a nonempty edge set (i.e. MOVE LifeSteps), where $E(v^a, u^a)=\{(v,u)|v \in [v],\ u \in [u],\ (v,u) \in E\}$, and*
- *M^a is the set of aggregate measures, each of which is computed by applying an aggregate function $A(\cdot)$ on the measure values $M_i(v)$, $v \in [v]$ and $M_i(e)$, $e \in E(v^a, u^a)$, respectively, $1 \le i \le n$.*

Note that the set of measures may be different for vertices and edges depending on the application, while the *A(•)* aggregate function may be different (e.g. *COUNT(•)* or *AVERAGE(•)*) for each of the measures M_i. Of course, in case of spatial or spatio-temporal measures will probably require more sophisticated aggregate functions. The key issue is for the aggregate function not to be holistic, so as the super-aggregates cannot be computed from sub-aggregates, even if we employ auxiliary measures [28]. Note also that the unification process U_V (U_E) operates on the spatio-temporal and semantic properties of the non-*Move* type (*Move* type, respectively) *LifeSteps*.

Definition 11 (semantic mobility cube): *Given a semantic mobility network N = (V, E, M) and a set of dimensions D = {D_1, D_2, …, D_m} with their corresponding hierarchies, the semantic mobility cube is the lattice of the aggregate semantic mobility networks produced by all possible aggregations in D.*

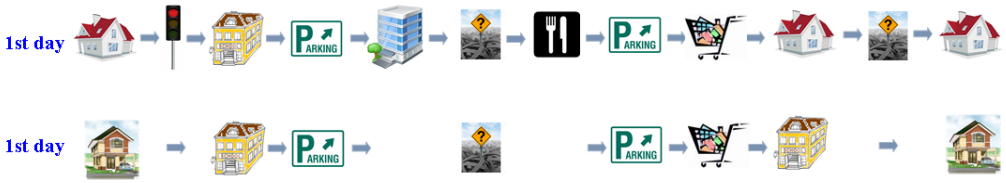
The above definition implies that given a SMN *N*, each aggregation D^a of *D*, called a *semantic mobility cuboid*, is itself a graph. To the best of our knowledge, this modeling approach is novel for the mobility domain, since it has been studied only for non-spatial, vertex-specific multidimensional networks of traditional datatypes [61].

For a deeper comprehension of the above definitions, in *Figure 13* we present another example demonstrating several novel operations/queries upon SMN. In particular, *Figure 13*(a) illustrates the sequence of the two 1-day semantic mobility timelines of Figure 11, while *Figure 13*(b) presents their aggregate SMN, according to Definition 10. This time, the aggregation actually stands as a merging operation upon the SMNs of the two users. What is more, in *Figure 13*(c) we depict their common SMN, relaxing the constraint that they should follow similar routes in order to be aggregated in the resulted SMN. The red sub-network in *Figure 13*(c) corresponds to their common SMN, where the space and time components of the two timelines have the same information.

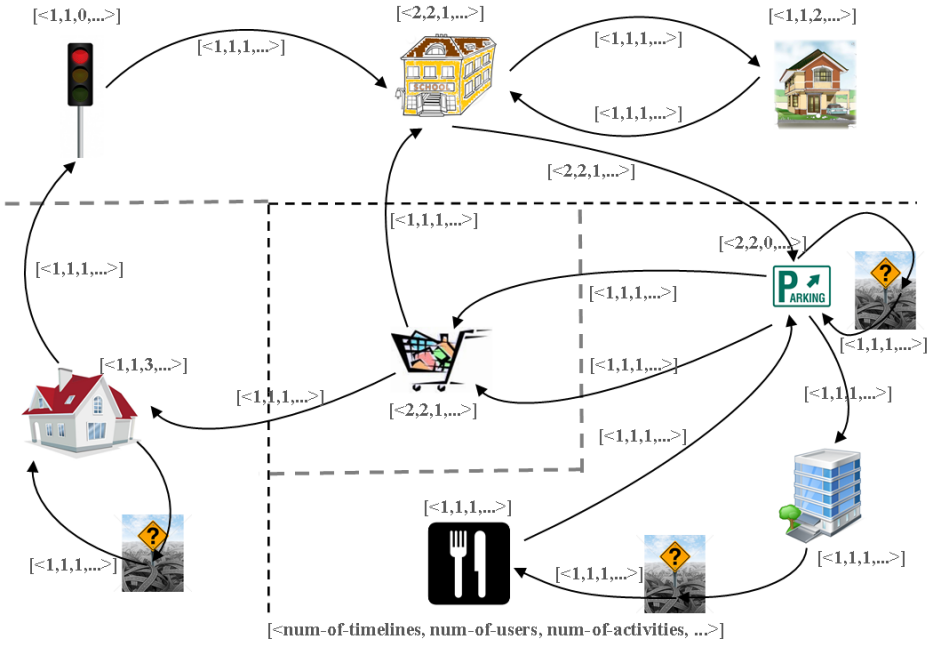
Continuing the example of *Figure 13*, in Figure 14 we adopt spatial hierarchies for interesting roll-up operations. Taking into account the spatial hierarchy illustrated in Figure 14(a) (implied by the black

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

(first level) and grey (second level) dotted lines in Figure 11), in Figure 14(b) we present the rolled-up network from “Business district” to “Shopping district”.

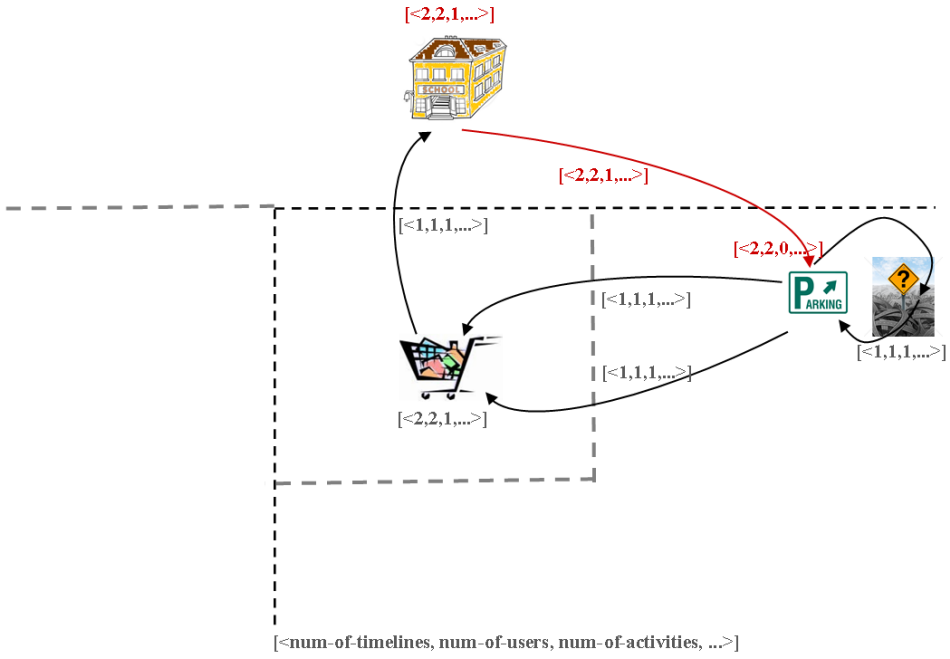


(a)



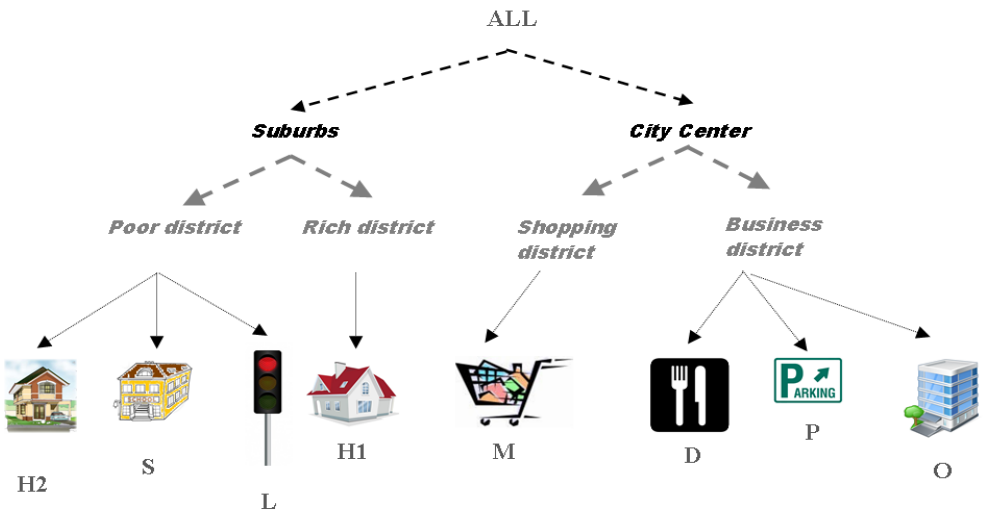
(b)

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833



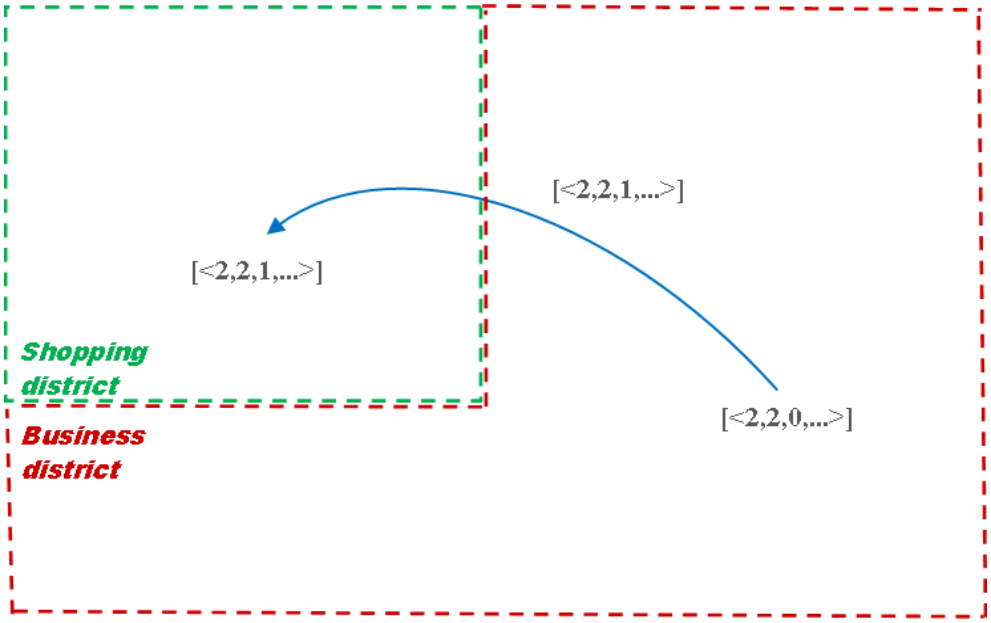
(c)

Figure 13: Querying Semantic Mobility Networks



(a)

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833



(b)

Figure 14: Performing OLAP upon Semantic Mobility Networks

It is also interesting to note that such example operations can be extended by query operands registered in the SMD for progressive analysis purposes. For instance, after we *"extract the mobility network of a user A during a period of time"*, we could *"restrict it at a particular region of interest"*, by using a range query. Assuming this network is at the base cuboid level, we could then join it with *"the aggregated (over a period of time) network of a set of users B (A's friends and co-workers according to a social network)"*, as such *B* resides at a higher level in the lattice. The join result (which obviously is a novel cross-SMN operation) could identify the network of A wherein she performs same activities by following similar (or dissimilar) routes, or if a different representation is required, the common and/or frequent places, or paths. Even more challenging could be a query like the following: *"Which timelines (or raw trajectories) conform to a given SMN, at any granularity level?"*, which is a cross-SMD/MOD-SMN operation.

Realization of the Hermes^{sem} MOD/SMD framework

Towards the realization of the architecture proposed in Figure 1, natural questions that arise include the following: (i) *how would a SMD be developed to provide efficiency in storage and querying*? and (ii) *how would a SMN be developed and what is OLAP upon SMN*?

Regarding the first question, what is required is the design of efficient storage schemes, like [30][42] did for trajectory databases. We followed the object-relational (OR) approach and extended the type system of Hermes MOD engine [42] and its associated query language to support mobility timelines. In

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

detail, we define a *Timeline* datatype (and a *LifeStep* datatype as a subtype of the former), upon which we register a rich palette of object methods. The resulted query language that is simple SQL extended with methods and operators appropriate for such complex (spatial + temporal + textual) objects, is successfully applied to a unique, proprietary real dataset that includes *synchronized* GPS traces and diaries, which is available in DATASIM. The presentation of the query language is presented in [13], where we provide details of the application interface of the query language and, even more important, we demonstrate the latter by applying a real-world case study upon this dataset. Another case study where we have applied the developed framework is available in Chap. 12 in [9].

The above extension would be of limited usage if the query operators do not have index support on such complex objects. We propose a solution, exploiting on spatio-textual indexing structures already proposed in the field of geographic information retrieval [56]. Of course, those structures ignore the time dimension, as such suffering from the same shortcomings that R-trees have when indexing mobility data. A different alternative would be to use a data structure tailored for mobility data, such as the well-known TB-tree [45], accompanied with a text index (e.g. inverted file).

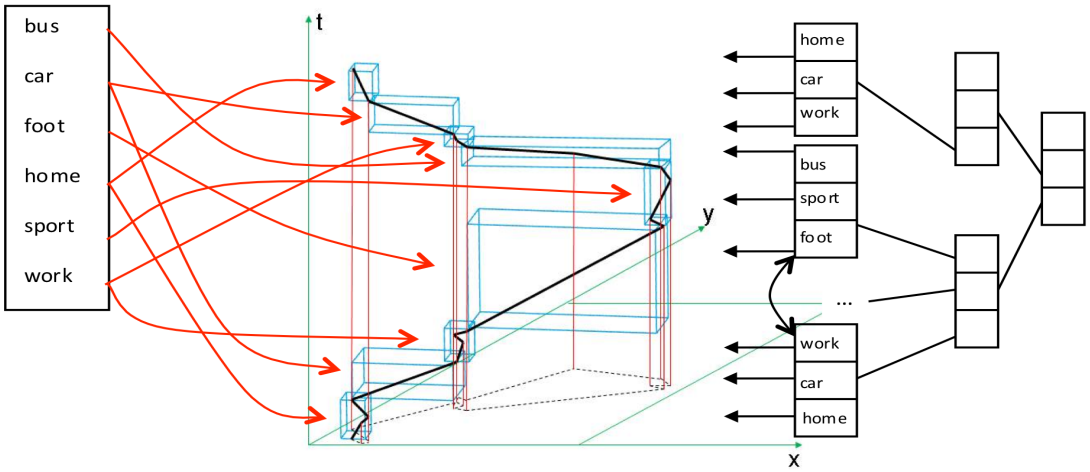


Figure 15: Hybrid indexing of “spatial + temporal + textual” information using TB-trees and inverted files

The intuition for choosing TB-tree is its trajectory preservation property, so as in each leaf node we maintain only those LifeSteps that belong to the same semantic mobility timeline. The idea of such a hybrid access method for semantic trajectories is illustrated in Figure 15, where we exhibit how the LifeSteps of a single Timeline are indexed. The Timeline corresponds to a user’s trajectory consisting of nine LifeSteps (whose MBBs in the 3D space are shown in the middle of Figure 6). The right part of Figure 15 illustrates a TB-tree that is constructed by these MBBs. Interestingly, although the tree is built upon the spatio-temporal component, it maintains the textual information for each LifeStep on the corresponding entry for that LifeStep on the leaf node (for clarity, only the tags of the LifeSteps are

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

illustrated in the leaves of the tree). Such choice for indexing allows for combined queries with combined spatio-temporal and textual constraints.

The above discussion prescribes that queries' resolution can take advantage of a filtering step on the spatio-temporal constraints of the query. Then a refinement can take place by using the textual constraints (by following the linked list at the leaf level). On the same line, we could utilize an inverted file built on the textual terms of the tags of the LifeSteps (see the left part of Figure 15), so as to enable the reverse filter-refinement process (i.e., first, filter by textual terms using the inverted file and, then, refine by the spatio-temporal constraints using the TB-tree). Obviously, the question of which of the two alternative filter-refinement strategies one should follow is an interesting query optimization problem.

Regarding the second question set in the beginning of the section, what is implied is the definition of aggregations over mobility timelines (see, for instance, the SMD data cube illustrated in Figure 1), a problem very related to the graph cube problem discussed in [61]. Recalling the idea of trajectory DW (TDW) [38], our proposal for developing a SMN is as a constellation scheme consisting of at least five dimensions (i.e. space, time, user profile, STOP-type activity, MOVE-type activity) and two fact tables (i.e. STOPS-fact and MOVES-fact). Intuitively, this approach allows the support of the following kinds of analysis, extended examples of which can be found in [13]:

- *STOPS-Fact*: who made a stop? when and where? what did she do during her stop?
- *MOVES-Fact*: who made a movement? when and from/to where? How did she move and what did she do during her motion?

In Figure 16, we provide a relational scheme of an effective modeling of SMN, where the measures of the fact tables correspond to the weight vectors of a SMN. We adopt the relational model to develop our proposal so as to be fully compatible with our approach of extending a real DBMS with semantic-aware functionality over mobility data. This actually permits the use of the extended query language previously mentioned [13] when filling in the SMN during the ETL process.

We argue that deriving a SMN from a SMD (i.e. the ETL process given the above choice) is a computational challenging task. For instance, a design decision could be that the SMN is built at a very refined spatial granularity (namely, at the level of the POIs and not at the region level, as in the case of TDW), while measures like the number of moving objects performing specific activities is also subject to the *distinct count problem* [38].

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

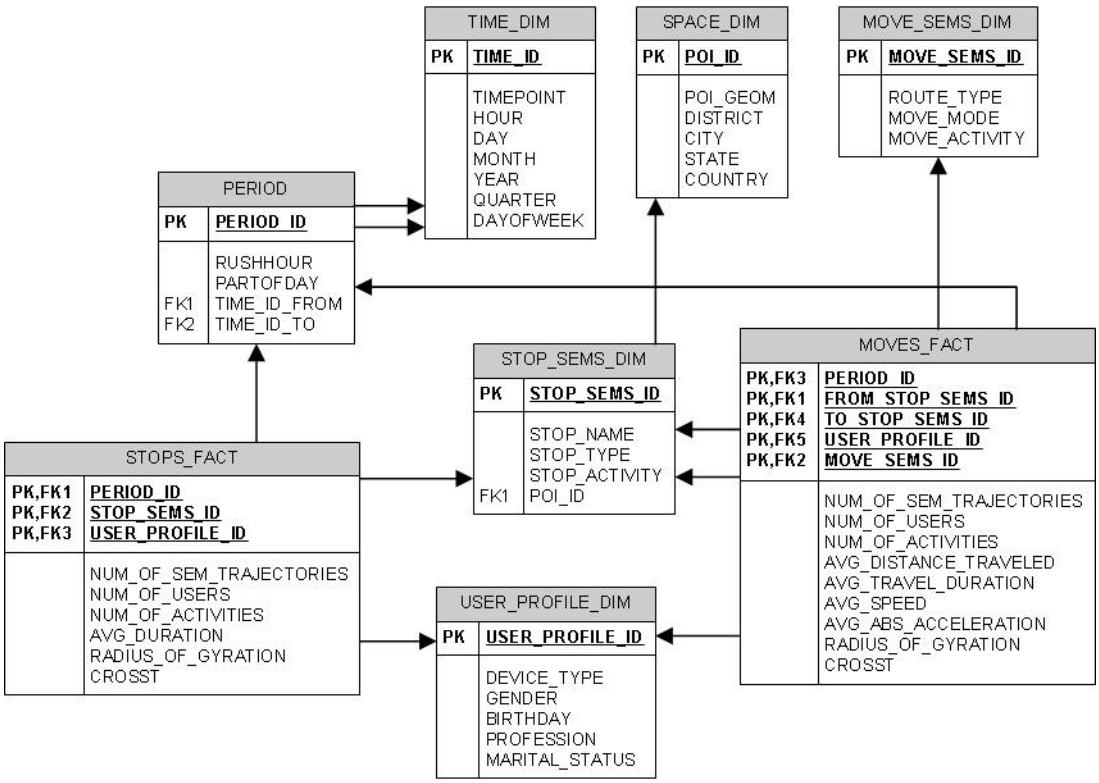


Figure 16: A data cube approach for SMN; a constellation scheme consisting of two fact and five dimensional tables

Summarizing, we extended related work on semantic trajectories [41] by introducing the novel concepts of LifeSteps, mobility timelines, and semantic mobility networks, which face trajectories of moving objects under a new perspective. Also, we presented the architecture of an integrated MOD/SMD framework and a list of interesting query types. What is more, we realized this framework and we presented its key components that implicitly highlight interesting research topics that require further investigation.

Although semantic mobility database management and analysis is still in its infancy, this paper foresees some research directions that will very soon be considered as real requirements. Summarizing, we outlined the following challenges as a minimum research agenda for the near future: (i) efficient storage and querying of data whose type from an abstract point of view may be considered as complex (spatial + temporal + textual) objects; (ii) efficient analytical processing upon such objects and SMN derived from them.

Building upon the latter, the issue of efficient and effective data mining operations over SMD and SMN seems to be even more challenging. This implies, on the one hand, extending existing trajectory data

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

mining algorithms in the new domain or even inventing novel ones and, on the other hand, incorporating them into the query language.

Baquara: A Holistic Ontological Framework for Movement Analysis using Linked Data

Another activity to meet the strategic objective for the semantic awareness of the data manager was to explore the narrowing of the gap between the spatio-temporal aspects and the semantics involved that hinders trajectory analysis, benefiting from the growing collections of linked data, with well-defined and widely agreed semantics, already available on the Web. The result of this effort was Baquara [5], an ontology with rich constructs, associated with a system architecture and an approach to narrow this gap.

The motivation starts from the fact that a sample point from a raw trajectory can be associated with keywords that carry information about the movement in that location. Lots of information can be extracted from such data, with a myriad of applications. For instance, information extraction methods can find *episodes* in raw trajectory data, i.e., maximal trajectory segments complying with a predicate [41]. However, these methods usually consider only the spatio-temporal component of trajectories, and do not address the specific content of episodes, whereas episodes can be thought of as segments of trajectories that carry specific semantics (e.g., a stop to take a picture of a monument or to take part in a sports event).

Effective analysis of movement must consider the semantics of the trajectories and the reality in which they occur. As already discussed, some conceptual models have been proposed for semantic trajectories analysis, later generalized for higher-level analysis [21] (e.g., goal, behavior, transportation means). Ontologies have also been proposed to support reasoning on knowledge bases describing trajectories [57][48]. However, these works do not address the automatic enrichment of trajectories with semantically precise information about specific places (e.g., restaurants, hotels, touristic spots), events (e.g., sport events, cultural events), and other relevant entities of the open dynamic world in which trajectories occur. In this work, a *semantic trajectory* is a sequence of episodes linked to specific concepts and/or instances via ontological relationships that can describe their precise semantics. Such semantic enrichment requires lots of continuously updated information, with well-defined and widely agreed semantics.

Baquara consists of an ontology (Figure 17) with an associated architecture (Figure 18) and an approach to enable semantic enrichment and analysis of trajectories with vast and growing collections of linked data available in the Web. The proposed ontology has a rich repertoire of constructs to semantically describe trajectories and their relevant episodes with linked data. Baquara plays the role of a conceptual bridge between movement analysis and the semantic Web, by allowing movement data and associated knowledge to be connected and queried together.

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

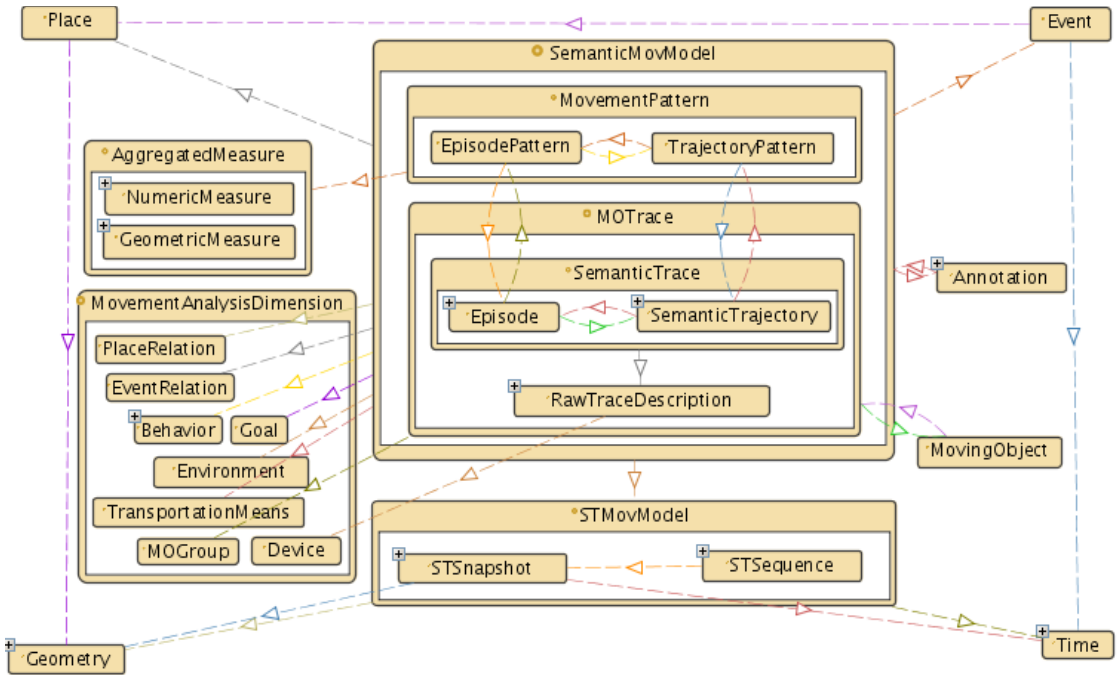


Figure 17: An overview of the Baquara ontology

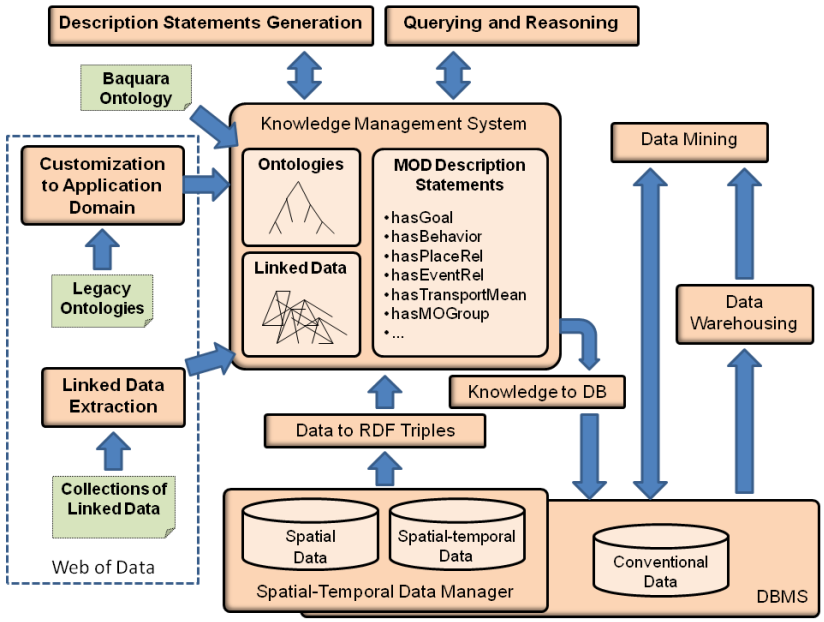


Figure 18: An architecture for semantic enrichment and analysis of MoD with linked data

Our approach takes as input raw movement data associated with conventional data that may not have precise semantics (e.g., tag "Rio" may refer to a city, a state, or even a nightclub, among other possibilities). The linked data that help to describe and analyze the trajectories are selected according

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

to the spatio-temporal scope of the movement to be analyzed, and the application domain (traffic analysis, tourism, emergency planning, etc.). Several methods can be used to find connections of movement data with linked data, including lexical and spatio-temporal matching (e.g., tag “Rio de Janeiro” associated with an episode occurring inside that city). After relevant episodes have been extracted and semantically enriched with linked data, powerful queries can be executed in the resulting knowledge base. Such queries could be from very specific, e.g., “Select the trajectories with at least one episode related to a touristic place called *Corcovado* in *Rio de Janeiro* city, even though the episode happens up to 10 kilometers away from *Corcovado*”, or abstract enough to only refer to concepts, e.g., “Select the trajectories that have a stop related to any *sport event*”. Although this is also related with the goals of WP2, in WP1 the proposed approach enables queries that refer to specific entities and classes taken from linked open data sources. For more details about this activity the reader is referred to [5].

Mobwarehouse: A semantic approach for mobility analysis with a Trajectory Data Warehouse

Substantial research has been conducted on providing methods and prototype systems for enriching trajectories with domain knowledge like points of interest visited by the users, transportation means, user activities and annotations. The model CONSTAnT [21] proposes a conceptual model for semantic trajectories, but with no specific reference to a data warehouse model. Nevertheless, CONSTAnT model inspired our current work in combining spatio-temporal and semantic aspects in a general concept of semantic trajectory.

Given this context, we propose a comprehensive TDW model for mobility called Mob-Warehouse, enriching trajectory data with domain knowledge. In particular, the model is based on the so called 5W1H (Who, Where, When, What, Why, How) framework. This is a well-known approach for getting the complete story on a subject, often mentioned in journalism, research, and police investigations. In our case, we intend to use this framework for specifying contextual information on trajectories and analyze the different aspects of the “mobility story” that the user “is writing” with his/her tracks. It will also guide the specification of the ETL process, which integrates trajectory data and domain application data. We developed a prototype implementation of the model and we experiment it in a case study consisting of a large dataset of car trajectories.

As already discussed, raw trajectories are, by nature, semantically poor and they have to be enriched with domain knowledge in order to achieve better understanding of moving objects behavior.

The 5W1H model

The basic idea of the 5W1H approach is to apply six narrative questions of Who, What, When, Where, Why and How to provide a consistent amount of understanding of the context of a circumstance.

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

We describe an object (Who) moving by a transportation means and/or having a certain behavior (How), performing an activity (What), for a certain reason (Why), at a given time (When) and place (Where).

Using this narrative approach, we may increase the level of semantic information into our model allowing to perform more meaningful queries about moving object habits. Below, we discuss the correlation of each question with trajectory features:

Who: This addresses the identification of a moving object, which is easily answered in case all objects are identified by the tracking system.

Where: This concerns the place where the trajectory point is located. Having the georeferenced location of each trajectory point, we may associate the latitude and longitude with a set of points of interest.

When: This question refers to the time extent related with trajectory points. This question is necessary to associate sampled points with specific calendar events, week periods, at different levels of details.

What: This question refers to what a moving object is doing, what task it is trying to perform or achieve. Clearly, answering this question is a challenge since we should infer from each trajectory point the corresponding activity. This process could be facilitated using the information provided by other questions.

Why: This represents the motivation for traveling. This is an issue since this is so deeply rooted into the moving entity intent.

How: This question identifies both the way the object is moving, like the transportation means, and the behavior like belonging to a traffic congestion. Answering to this question could be challenging: when trajectories use multi- modal transportation means, identifying the transportation means could be far to be trivial. Moreover, mining algorithms have to be used to extract meaningful behavior: trajectories can be associated with one or more patterns depending on the fact the trajectory entails or satisfies the behavior.

The TDW conceptual model

Following the 5W1H Model we define a TDW conceptual model with six dimensions, as illustrated in Fig. 1. First, two dimensions represent space and time and they correspond respectively to the Where and When questions of the 5W1H Model.

At the base granularity it represents a single sample (id, x, y, t) belonging to the trajectory identified by id. The hierarchy having Sample as a root mixes together semantic and geometric features. A sample belongs to an episode, which can be classified according to its Type (e.g., a stop or a move) and it is grouped into a Trajectory. Each Trajectory is associated not only with the Moving object but also to a Goal, which is the main objective of such a trajectory. This dimension allows one to model Who is performing the action (the moving object) and the attribute goal answers the question Why. A fourth dimension, called Activity, states the activity the object is doing in a certain sample. This allows one to

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

describe in a very detailed manner What is going on at the different samples of a trajectory. We can build a hierarchy of activities which classifies properly the variety of things an object can perform. Usually this hierarchy is application dependent hence in the general model it is not specified and should be instantiated case by case depending on the application requirements. Then the dimension Means of Transportation represents which transportation means the object is using for the movement. The last dimension, called Pattern, collects the patterns mined from the data under analysis. In this way we can directly relate trajectories to the patterns they belong to. An example of hierarchy on this dimension could be: each pattern is associated with its Type, such as cluster, frequent pattern, flock, and with a Semantic Pattern, which expresses the interpretation of such a pattern. For instance, in [2] to compute the movements of commuters in the city of Milan a clustering algorithm is applied to the trajectories of the moving objects. We can store these mined clusters in Mob-Warehouse as follows. Each cluster ci is represented as a row in the dimension table Patterns: the identifier of the cluster is the primary key, the attribute Semantic Pattern can state north-east commuters, to point out that the semantic interpretation of cluster ci is a group of people moving from North to East whereas the attribute Pattern type assumes cluster as value. The latter two dimensions express the concept of How the movement is performed.

The fact table stores measures concerning the samples of the trajectories.

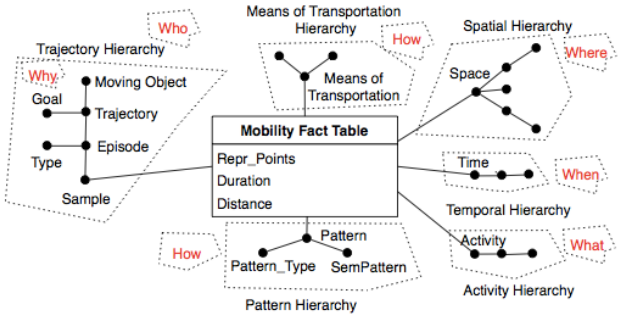


Figure 19: The MobWarehouse conceptual model

Experiments. A real case study using a trajectory dataset of people traveling by car in Milan (Italy), during one week in April 2007 has been performed. The dataset contains track of 16,946 cars and 48,906 trajectories for a total of 1,806,293 points. After the ETL has been performed, the system is able to answer semantically rich DW questions like:

1. Find the most frequent activity after home

```
SELECT ac.category, COUNT(*) FROM points_fact pf, traj_dim tr, activity_dim ac, stopsAtHome stops
 WHERE (join conditions) AND tr.trajectory = stops.trajectory
 AND tr.episode = stops.episode + 2 AND ac.category <> 'HOME' GROUP BY ac.category ORDER BY 2 DESC;
```

2. Find the users and their time spent traveling from home to work

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

```
SELECT traj_dim.trajectory, SUM(points_fact.duration) AS time    FROM points_fact, traj_dim,
activity_dim, time_dim    WHERE (join conditions) AND    ((traj_dim.episode_type = 'BEGIN' AND
activity_dim.category = 'HOME')
     OR (traj_dim.episode_type <> 'BEGIN'))    AND time_dim.minute <=
firstWorkStop(traj_dim.trajectory)
GROUP BY traj_dim.trajectory;
```

A Privacy skin for the Semantically-Enriched Data Manager

Regarding the issue of privacy, although a huge concern, as explained in the DOW, in DATASIM it is not a main research topic. More specifically, as prescribed also in D1.1, in addition to the approach of data anonymization by concurrent adherence to the privacy-by-design principles, we employ a more conservative approach by assuming that the majority of the information that is captured in the mobility dataset must remain private and that the data has to stay in-house to the hosting organization. Following this approach, in order to ensure privacy-aware sharing of in-house mobility data, a mechanism is necessary to control the information that is made available to external parties when they query the database, so that only non-sensitive information leaves the premises of the hosting organization. This section presents an overview of the privacy-aware 'skin' of Hermes^{sem}, called Hermes^{sem++}, for handling data privacy aspects. Further details of this approach can be found in [17].

Most of the proposed approaches for movement analysis are mainly focused on the raw spatiotemporal features of trajectories stemming from GPS devices. However, the increasing progress of mobile technology along with the interest of understanding human mobile behavior has led to the creation of more complex representations of trajectory data. Raw trajectory data are enriched with conceptual knowledge conducting more meaningful and comprehensible abstractions of mobility data. This alternative representation of trajectories may pose even greater privacy violation threats. Consider for example a malevolent user that is able to detect *places of interest* (POIs) where a moving object has stopped (e.g. living area, working buildings, and public places such as a mental health clinic, etc.). This additional knowledge that has gained allows the inference of personal sensitive information of this specific individual.

In general, the data privacy problem requires protecting the privacy of individuals from being re-identified by adversaries. Due to the sequential nature of mobility data, the discovery of quasi-identifiers is even more difficult in contrast to relational data. The dependence between consecutive points of a user's trajectory both in spatial and temporal dimension does not allow to determine a specific set of locations and the corresponding time intervals to be the quasi-identifiers. Sensitive locations vary between users since a random location of one user might be sensitive for another. Based on the assumption that the data publisher is not always aware of the adversary information, we consider every part of a semantic trajectory (either stop or move) as sensitive.

Moreover, an optimal trade between privacy and data utility should be achieved. On the one hand, data should be transformed in order to avoid re-identification attack. On the other hand, information loss should be controlled. As a result, it is necessary to quantify the risks of privacy violation while minimizing information loss by measuring the utility of the data after the transformation process.

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

In order to handle these more complex representations of data, we realize and extend the design principles of HERMES++ that were proposed in [43] by developing Hermes^{sem++}, a query engine for semantically-enriched trajectory data that allows subscribed end-users to gain restricted access to the database to accomplish various analysis tasks. Hermes^{sem++} can shield the trajectory database from potential attacks to user privacy, while supporting popular queries for mobility data analysis for both raw and semantic trajectories. Similarly to [43], our engine audits end-user queries and operates by retrieving real user trajectories from the database and generating carefully crafted, realistic fake trajectories that preserve the trend of the original data in order to reduce the confidence of attackers regarding users' identification from their real trajectories in the query result. However, unlike the privacy-engine that was proposed in [43], Hermes^{sem++} achieves to provide answers to popular queries for semantically-enriched trajectories (Q_1: *Find people that stop at the* ***Market*** *at 5pm)* and for semantic mobility patterns (*Find people starting their trajectory from* ***Home*** *between [8.00-8.30am], having a stop at a* ***Mall*** *between [5.00-8.00pm] and, then, have a stop at a* ***Restaurant*** *between [9.15-11.30pm])*, while protecting the sensitive semantic locations by ensuring that no sensitive locations that would lead to user identification are reported as part of the returned trajectories.

In more detail, given a database of in-house semantic trajectories and users' privacy profiles (i.e. users are able to specify which locations are sensitive), the engine provides answers to queries while guaranteeing that the identity of a user and the visited sensitive places will not be inferred with probability greater than a user-defined threshold. Moreover, we assume that a malevolent: (a) has gained access to the query engine and can pose queries in the available semantic trajectory database, and (b) may possess background knowledge (i.e. is aware of users' privacy profiles). Hermes^{sem++} can effectively protect user's privacy by blocking three types of re-identification attacks (*semantic location identification attack, sensitive location attack, sequential tracking attack*) that malevolent users may try to pursue in the original database.

Our framework works as follows (Figure 20). A user poses a query to the database. Then the auditing mechanism examines if a privacy violation attack has occurred w.r.t. to user's history. In such a case, the query is rejected and no answer is returned to the user. Otherwise, it proceeds by retrieving the set of real semantic trajectories that correspond to the answer. Based on the real answer set, the engine then generates a set of carefully crafted realistic semantic trajectories. The history is then updated for future convenience and the fake trajectories are returned as an answer to the user.

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

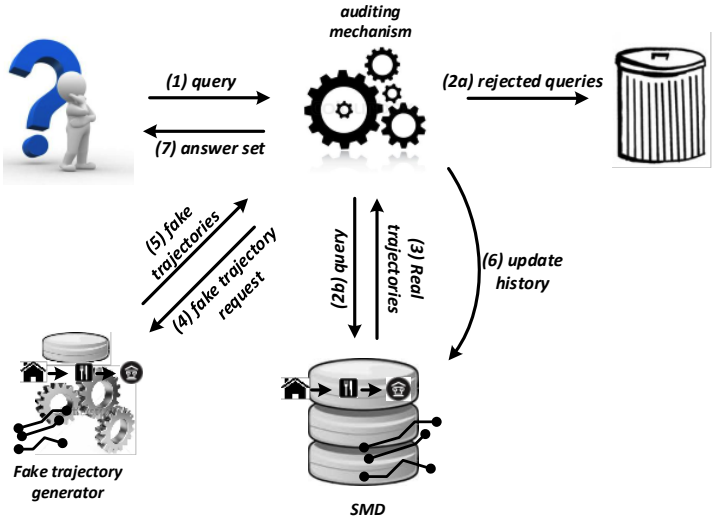


Figure 20: Workflow of Hermes^{sem++}

The malevolent user, given a set of trajectories answering the query posed to the semantically-enriched trajectory database tries to increase his confidence regarding the identity of a given user by performing a set of more focused overlapping or nearby queries and/or linking sensitive places related to the specific user. Regarding user protection perspective the main goal is to minimize re-identification probability. However, an optimal trade-off between minimizing the probability while maintaining data utility should be achieved. *k*-anonymity principle is adopted to control re-identification probability. In order to provide an answer to a query posed by a user, re-identification probability should not exceed the threshold $1/k$, where *k* is a user defined threshold. However, since we assume that a user has the ability to pose a set of queries, the corresponding answers are highly correlated. Due to this fact re-identification probability should take into account the history of the queries posed by the same user. Thus, the proposed auditing mechanism is able to handle the aforementioned privacy attacks while ensuring that the identity of a user will not be exposed with probability greater that a user defined threshold.

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

5. Meeting BIG data challenges for mobility data

In this line of research we investigated solutions to make the above-described infrastructure capable to deal with really BIG data. To meet this challenge, we investigated several computing paradigms as distributed and parallel DBMSs or MapReduce-based solutions and we first studied exhaustively the corresponding literature Chap. 10 in [9]. In the following section we appose the few related works as the subject has only very recently started to be explored in the literature of mobility data. Then the tactic we followed was not to try to design from scratch a solution that can tackle BIG volumes of mobility data, but to devise appropriate segmentation / partitioning methods [14][15] that can be exploited by off-the-self distributed solutions. In different words, the approach we followed is to partition / distribute mobility data in a smart way, so that we take advantage of the efficiency of the local computation nodes, by using the latter in off-the-self distributed and parallel solutions.

Moreover, we require few features of this partitioning method for mobility data:

- As we want our framework to be able to handle many different tasks equally well, e.g. we want to treat mining operations as first-class citizens, equally well with querying operators, the partitioning should be performed by a mining method too, actually the previously discussed S^2T-clustering method. The idea is simple and old: we follow a divide-and-conquer approach in that first we cluster by simultaneous partitioning data (divide step) and then, depending on the required task, we merge data or patterns (conquer step) to form the actual result.
- A second desired characteristic of the partitioning method is to have incremental characteristics. Assume, for example, the following scenario: Uers transmit their location to a central server *asynchronously* and in batch mode; from the server side, a Moving Object Database (MOD) system is responsible for organizing users' traces in terms of *trajectories*; for providing high quality services, the server executes extensive querying and mining processes on the trajectory database stored in the MOD engine. Unlike BIRCH and CURE for relational data, in the MOD domain, there is a lack of an efficient incremental clustering algorithm. The '*incremental*' characteristic is essential since updates in mobility database are frequent and clustering results can quickly degrade. This incremental clustering approach should also support PAUSE/RESUME operations due to the big volume of data to be handled.

The following section presents related work in the field, while the subsequent two provide insights the proposed sub-trajectory clustering method [14] and how we use it to introduce our incremental partitioning method [15] that is the base of solution to tackle BIG data. Interestingly with this approach we are in position to explore different computing paradigms. We do this by applying it to a distributed and parallel DBMS (i.e. PL/Proxy), as well as with a MapReduce-based solution (i.e. HadoopDB).

Related work

Devices such as mobile phones, tablets, automobiles and generally GPS-enabled devices produce trillions of bytes of information about their user's movement. This enormous amount of data,

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

concerning the mobility of individuals, being generated in our world the last decades has posed new challenges in the world of data management. Various methods and technologies have been developed and adapted in order to store, manage, analyze, and visualize the complex vast quantities of data in an efficient way. These methods and technologies originate from different fields such as computer science, applied mathematics, economics, and statistics. Therefore, an organization that aims to gain value from this kind of data has to espouse a flexible, multidisciplinary approach. Some of these technologies have been deployed specially for huge volumes of miscellaneous data, while others were evolved for smaller sets and diversity in data, but have been successfully adjusted so that they are applicable to very large and diverse portions of data.

Generally, the term “Big Data” refers to a collection of, usually unstructured, datasets so large, diverse and complex that is beyond the ability of typical database management tools to capture, store, manage, and analyze. The challenges being posed to the data management community by big data include capturing, storing, indexing, searching, sharing, analyzing and visualizing this kind of data.

As far as it concerns storing and managing big data there exist two main paradigms: parallel and distributed databases. The common field of both is that they are decentralized, meaning that they try to distribute data and/or computing power to multiple nodes over a network. In particular, a parallel database system pursues to increase performance through parallelization of various tasks, such as building indexes, loading data, evaluating and executing queries. Parallel databases may follow shared-disk, shared-memory or shared-nothing architecture, as illustrated in Figure 21.

The architecture of *shared-memory* systems consists of multiple CPUs and a shared memory, with the access of each CPU to the shared memory taking place over a common bus; in *shared-disk* systems, every node has one or more CPUs and its related memory (nodes do not share memory), while the communication takes place above a mutual high-speed bus, and every node has access to the shared disks; finally, in *shared-nothing* systems, each node is independent and self-sufficient in terms of memory and disk storage.

Depending on the adopted architecture, data may be stored in a distributed fashion or not and try to improve processing and input/output tasks by using multiple CPUs and disks in parallel. Moreover, parallel processing is divided in *pipeline parallelism*, where numerous computers perform one step each in a multi-step process, and *partition parallelism*, where numerous computers perform the same operation over different fragments of data.

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

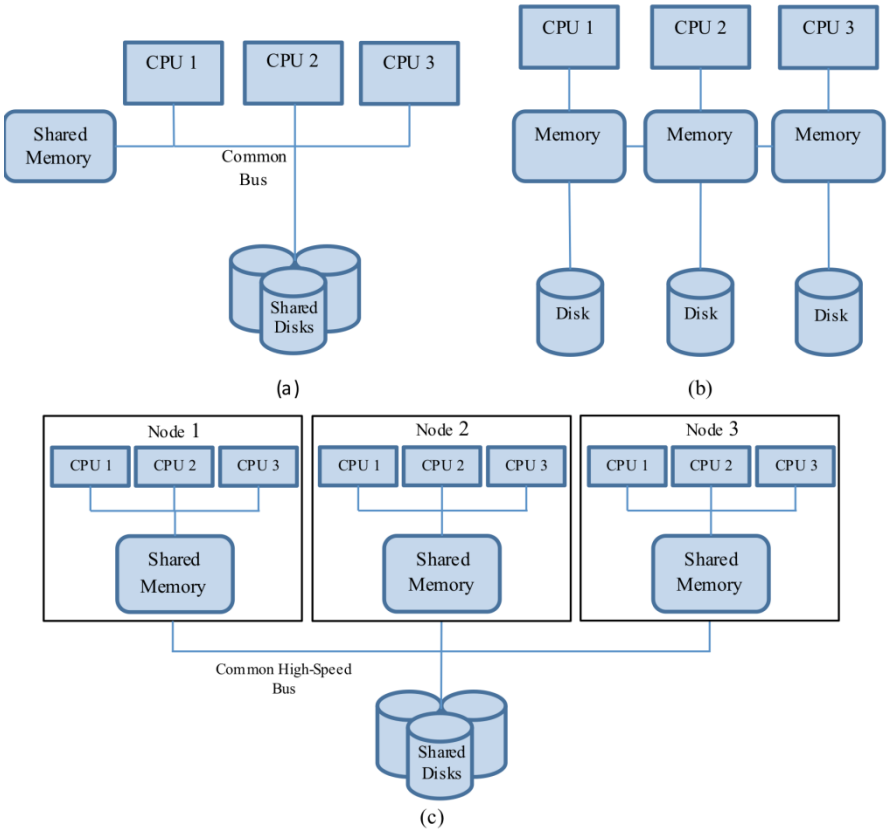


Figure 21: Alternative parallel database system architectures: (a) shared-memory; (b) shared-nothing; (c) shared-disk.

On the other hand, a distributed database system is a collection of logically related databases that co-operate in a transparent way. The term 'transparent' implies that users may access data irrespective of the database it is physically stored, as if all data were stored in a single database locally. This implies "location independence", i.e., the user is not aware of the location of the data, while it is possible for the data to move from one physical location to another transparently to the user. Distributed databases are usually deployed in a shared-nothing architecture, which leads to reduced communication overhead and better performance. Furthermore, the hazard of a failure on a site is eliminated, i.e., if a server fails, then the only part of the system that is affected is the corresponding local site and the rest of the system remains functional and available.

As one could easily observe, there is overlap between parallel and distributed databases. In fact, nowadays, the predominant technologies, which can cope with big data, are mostly classified as "shared nothing". Cases of such success stories include Google File System, MapReduce, Hadoop, Big Table, HBase, Cassandra, etc.

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

One of the predominant technologies, as far as it concerns Big Data, is MapReduce. MapReduce is a model of distributed programming created in order to process a particularly large volume of data, which are stored in different sites. This model is applicable to a wide range of use cases and is organized in a two-step Map and Reduce process. *Map* takes as input a set of (*key*, *value*) pairs and produces a group of *intermediate* (*key*, *value*) pairs. All the intermediate values associated with the same key are grouped together and go through to the *Reduce* function. In turn, *Reduce* receives an intermediate key and a collection of values for that key and it groups together these values with the intention to compose a probably reduced set of values. The transitional values are usually fed to reduce function through an iterator, which allows manipulating immense lists of values that cannot fit in memory.

The actual operation of the MapReduce scheme is shown in Figure 22 and performs as follows:

1. The input files are split into *M* pieces and several copies of the program are initialized on a set of machines which belong on a cluster.
2. One of these machines is set to be the *master*, while the rest are set to be *workers*. The master assigns either *Map* or *Reduce* tasks to the idle workers. There are *M* map and *R* reduce operations, and each worker is assigned a *Map* or *Reduce* task.
3. Each worker, which has been assigned a *Map* task, receives as input the corresponding input split and extracts from it the (*key*, *value*) pairs. Subsequently, these pairs are delivered to the *Map* function where the intermediate (*key*, *value*) pairs are emitted and buffered in the memory.
4. The local disk is divided into *R* partitions, which are defined by the partitioning function, and at regular time intervals some of the intermediate (*key*, *value*) pairs are written to it. The position on the disk, that each pair is written, is transmitted to the master node, thus making it responsible for passing these locations to workers that have been assigned a *Reduce* task.
5. As long as a *Reduce* worker becomes aware of the positions of the intermediate (*key*, *value*) pairs, connects to the corresponding *Map* worker through remote procedure calls and reads the data. After the *Reduce* worker has read all of the data, then it sorts them by the key. In this way, the same keys are grouped together.
6. Subsequently, the *Reduce* worker goes through the ordered intermediate (*key*, *value*) pairs and for every distinct key the respective (*key*, *value*) pair is fed to the *Reduce* function, while the outcome is appended to an output file for this reduce partition.
7. When all the *Map* and *Reduce* tasks have been performed, the master returns the control to the user program.

More details about technologies focused on addressing the Big Data problem can be found in Chap. 10 in [9].

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

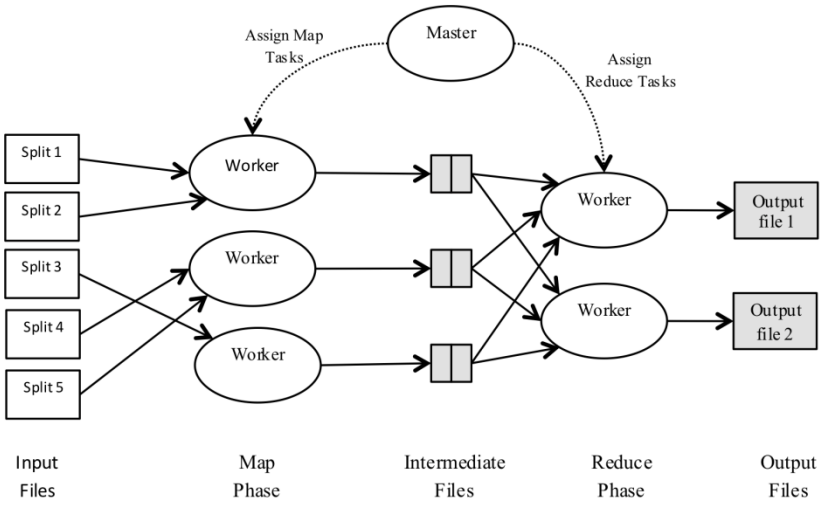


Figure 22: The MapReduce programming model.

An interesting instantiation of the Big Data Management problem is the manipulation of Mobility Data. Big mobility data management deals with amounts of data so huge that the traditional database systems are unable to handle. For this reason, the approaches for handling this kind of data involve non-traditional, distributed and parallel systems.

Mobility data applications can be classified into two categories: historical and real-time. The management of historical data implies that the data are stored in a single or multiple sites in an archival mode. Historical data are used for offline querying, analysis and extraction of useful knowledge. As such, the respective applications try to optimize querying over the entire dataset. An optimization process could involve indexing mechanisms, storing the data in a distributed way, etc. Usually, this process is application specific, meaning that depending on the queries that an application desires to apply over the data (e.g. range, NN queries, etc.), it tunes its indexing mechanisms and storage infrastructure in order to efficiently answer the queries. On the contrary, real-time mobility data get updated continuously (or in batch mode) with the current location of the objects. At the same time, the past locations of each object should be maintained for querying and analysis purposes. Based on the assumption that past locations of the objects are less likely to be retrieved, the effort is being focused on managing the recent and current versions of the data in such a way that they can be efficiently updated and queried.

Based on the literature, purely distributed systems, try to tackle real-time big mobility data. On the contrary, there are few efforts to deal with historical big mobility data by utilizing the MapReduce framework.

As far as it concerns spatiotemporal data services that necessitate the vast execution of update operations in order to keep the data up-to-date, there have been recently proposed an approach based on the MapReduce paradigm. More specifically, the so-called MOIST (Moving Object Indexer

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

with School Tracking) [34] the proposed architecture employs a shared nothing infrastructure, so as to store and index spatiotemporal data. This infrastructure consists of commodity machines connected through high speed networking devices, where each node has independent processing unit and storage devices.

The proposed framework utilizes a Master Site, multiple Storage Sites and a high speed networking device. The primary role of the Master Site is to index the recent history and the current version of the data. To achieve this, a Multi Version R-Tree (MVR Tree) is employed. By this, the most recent versions of the data are kept at the Master Site and the rest of them (the older versions) are distributed to other sites. Additionally, in order to guarantee that very high update rates can be achieved, the MVR Tree is kept in-memory. The Master site also plays the role of the Coordinator whose primary objective is load balancing in order for the data to be distributed in an intelligent way through the cluster of workstations and thus increasing concurrent query processing rates. The main responsibility of the Storage Sites is to use their abundant resources to archive the past versions of the data in their secondary storage devices and to handle efficiently data resident to them. The indexing mechanism deployed for indexing data stored in the Storage Sites is R-Trees.

The main goal in this approach is to reduce the update latency and to support efficient history queries. The four key components of MOIST are:

1. Key Value Model. It is the cornerstone of MOIST. Big table is used in order to store the Location Table, Spatial Index Table and Affiliation Table.
2. Spatial Indexer. Google S2Cell indexer is utilized in order to divide the space into cells of different resolutions. A space filling curve makes the cell efficient for spatial indexing.
3. Object School. MOIST clusters data into object schools, consisting of objects that have spatial proximity and similar velocities. In this way, redundant updates can be reduced and consequently the update latency can be limited.
4. Aged Data Archiving. A parallel ping pong scheme is employed in order to flush history data onto disks.

As already mentioned, the basic structures of MOIST are the Location Table and the Affiliation Table (Figure 23). The Location Table stores the location and velocity information for each object, in such a way that queries on an individual object can be answered efficiently. The Affiliation Table is used to keep track of object schools (OSes), by keeping information about the mapping for each cluster and is keyed by the behavior (i.e., the leader's speed) of that cluster. The data are index with S2Cell indexer. The space is divided into a number of cells of different resolutions and a space filling curve (Hilbert Curve) is constructed by linking all cells in a space by the sequential order of their spatial indexes. MOIST groups objects nearby to each other and with similar moving behavior into one school. This is based on the observation that objects close to each other often move with similar speed and with analogous trajectories. In order to keep track of an Object School, MOIST keeps track of the leader object only, and records the distance between the follower objects and the leader. OSes are

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

preserved and transformed when an update arrives. An object leaves its OS when its distance to its leader surpasses a given threshold. After an object departs an OS, it becomes a leader of a new OS. OSes are merged periodically using a quick cluster method. Finally, aged data should be flushed onto disk so that the history of the moving objects can be analyzed for extraction of useful information. A naive approach for achieving this is to move an updated object location onto the disk before a new update arrives. This can lead to a large number of disk IOs and consequently to a latency penalty. To reduce this latency overhead, a double-buffering (or Ping-Pong buffering) scheme can be utilized. While updates are occurring on one memory buffer, another one is moved onto the disk. What should be ensured in this scheme is that the time required for flushing aged data from the buffer onto the disk is less than the time required to fill the other buffer in memory.

Location Table

ID	Location
4	…
6	…

Spatial Index Table

Spatial Index	ID
…	6
…	4

Affiliation Table

ID	L/F	Follower Info
2	F-4	
4	L	2(4→2),7(4→7)
5	F-6	
6	L	5(6→5),9(6→9)
7	F-4	
9	F-6	

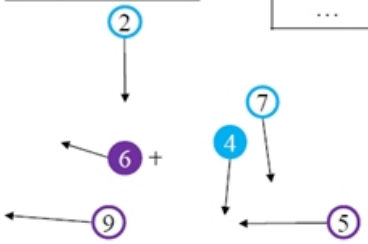


Figure 23: The content of Location, Spatial Index and Affiliation tables in MOIST for six objects organized in two object schools.

Regarding offline data analytics on Big Mobility Data an interesting approach that employs Hadoop is proposed in [53]. CloST proposes a way for storing big spatiotemporal data using the Map-Reduce framework so as to optimize two types of queries:

1. Single-object spatiotemporal range queries Q(I, S, T)
2. All-object spatiotemporal range queries Q(S, T)

Where S is the spatial range, T is the temporal range and I is the object id. CloST stores spatiotemporal data in tables, in the form of (Oid, Loc, Time, A1, . . . , An). The first three attributes are the core attributes where Oid is the object id, Loc is the spatial point, and Time is the timestamp.

To enable the efficient processing of both types of queries, a three-level hierarchical partitioning is employed, shown in Figure 24. At the first level the data are divided into a number of partitions according to hash values of object ids and coarse ranges of time. At the second level each partition

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

derived from the first level is divided into a set of partitions according to a spatial index on the location attribute. Finally, at the third level the actual data are found. Here level 1 and level 2 serve as indexes for level 3.

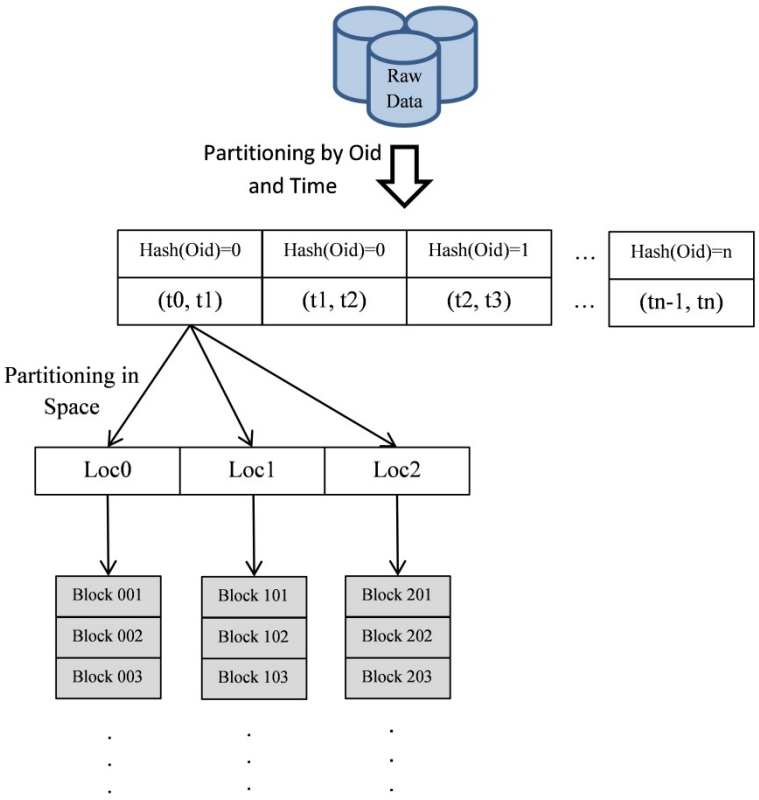


Figure 24: Hierarchical partitioning in CloST.

In order to achieve its goal, given the fact that Hadoop is employed, CloST devises a space efficient file format to actually store data in HDFS. Records are grouped by object id and each group is stored in a file section. An in-block index is placed at the beginning of each file in order to map each object into the corresponding position inside the file section. Then, for each section, records are sorted by time and then organized in a column-store fashion. To improve the storage efficiency, CloST compresses data first at column-level where numeric values are encoded by either delta encoding or running-length encoding. Afterwards, data are compressed at section-level where all data inside a section are compressed once more using gzip to further decrease the data size.

Furthermore, a Metadata table exists, containing information about mappings from buckets (level 1) to regions (level 2) and from regions to blocks (level3). This table is stored together with the data, under the same directory, which makes it easy to automatically replicate and distribute it along with the actual data. Finally, for a single object query, the Metadata table is used together with the in-block

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

indexes in or-der to locate the desired data. For an all-object query, the Map-Reduce framework is employed in order to execute the query in parallel. Primarily, all the blocks that intersect the spatiotemporal range are retrieved. Then each Map task sequentially scans the corresponding block file and assembles the records. Then it filters each record by S and T in order to output the final result. Finally, in order to optimize the blocks size, CloST retains a query log of the recently posed queries and periodically tunes both spatial and temporal partitioning.

A hybrid approach where not only historical data are treated by the Map-Reduce framework but also updates of moving objects can be handled efficiently is proposed in PRADASE (Query Processing of Massive TRAjectory DAta baSEd on MapReduce) [36] is based on a GFS-style storage which supports only appending data. The master node is in charge for key to (data) nodes mapping and lookup (via hashing). So, if the key is given, the corresponding data can be returned automatically. Many data intensive computational tasks can be simplified due to the fact that the execution engine is built on Map-Reduce. It is appropriate for large-scale distributed systems (i.e. GFS and Hadoop), since interactions among data on different nodes happens rarely. Here, the queries under optimization are

1. Spatiotemporal range query
2. Trajectory based query

The process of execution is:

1. Splitting input data into several pieces
2. Executing sub-task in each piece of data
3. Grouping all intermediate values with the same key
4. Reduction of each group.

More specifically, as shown in Figure 25, when new trajectory data is imported into the system, index module will use PMI creator (Partition based Multilevel Index which is proposed to deal with spatiotemporal range queries) and OII creator (Object Inverted Index which can facilitates trajectory based queries) to form two kinds of indices accordingly. After constructing the indices, all data is transmitted to storage module. When queries are posed, the results will be returned rapidly by invoking index module.

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

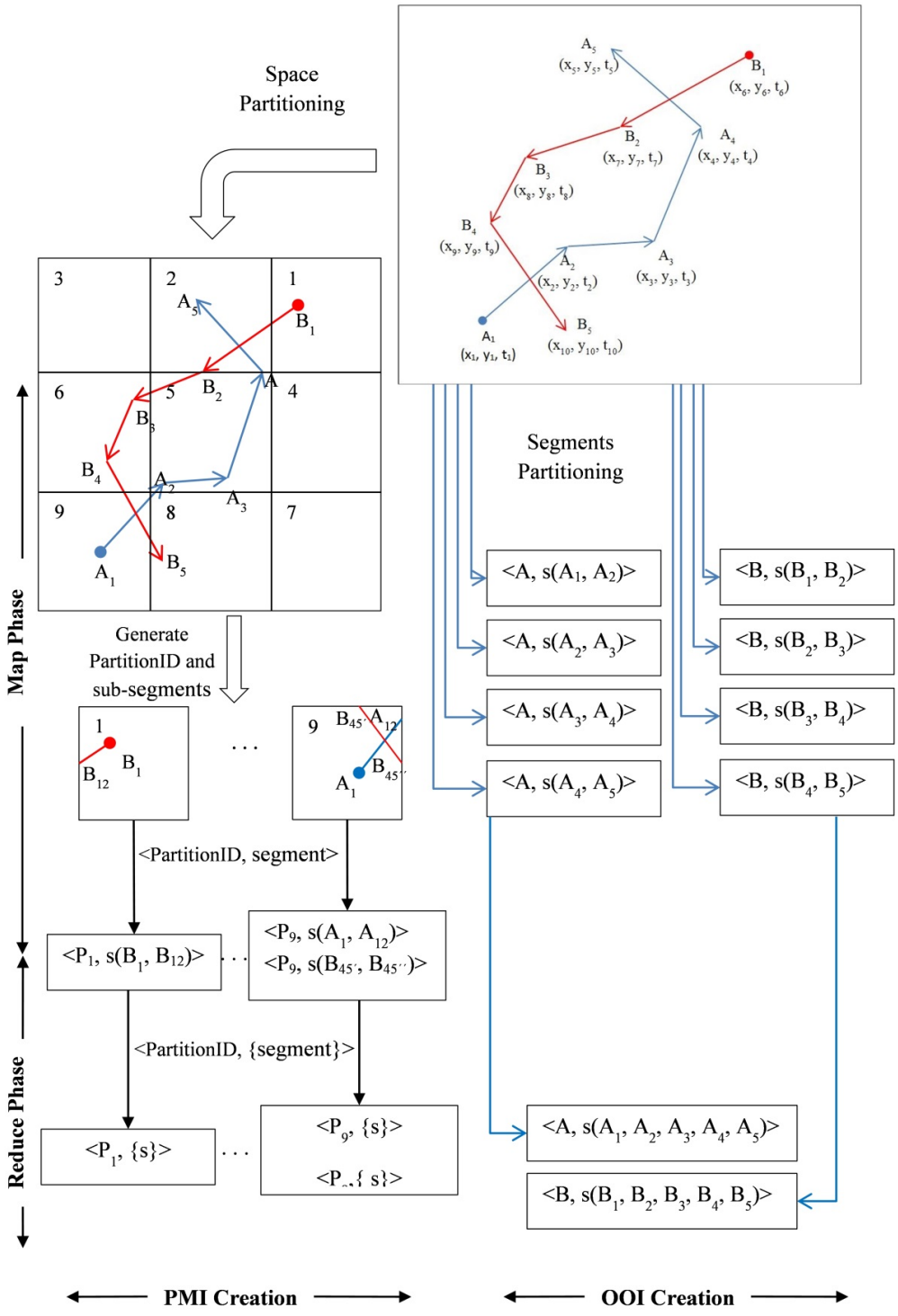


Figure 25: PMI and OII based Trajectory Data Insertion.

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

Regarding the query module, as far as it concerns the spatiotemporal range query, the spatial extent of the range is checked against the partition of PMI. As shown in Figure 26, for trajectory-based query, given an object id, the corresponding address of storage can be located directly by employing the OOI. For any given range query, the PMI index can speed up querying. In map phase, each data node generates a candidate partition set which consists of all potential partitions stored locally and the corresponding sub queries. Then, each data node reads the data, executes the sub queries and outputs the results for each sub query in the form of <*objectID*, {*segment*}>. After grouping by *objectID*, the reduce phase groups the sub results with the same *objectID* and executes a join operation for each object's all sub segments and finally returns the results <*objectID*, {*trajectory segments*}>. Furthermore, the OII index speeds up trajectory-based queries. Given any object ID, the system can locate where the data are stored and read historical trajectories of the object.

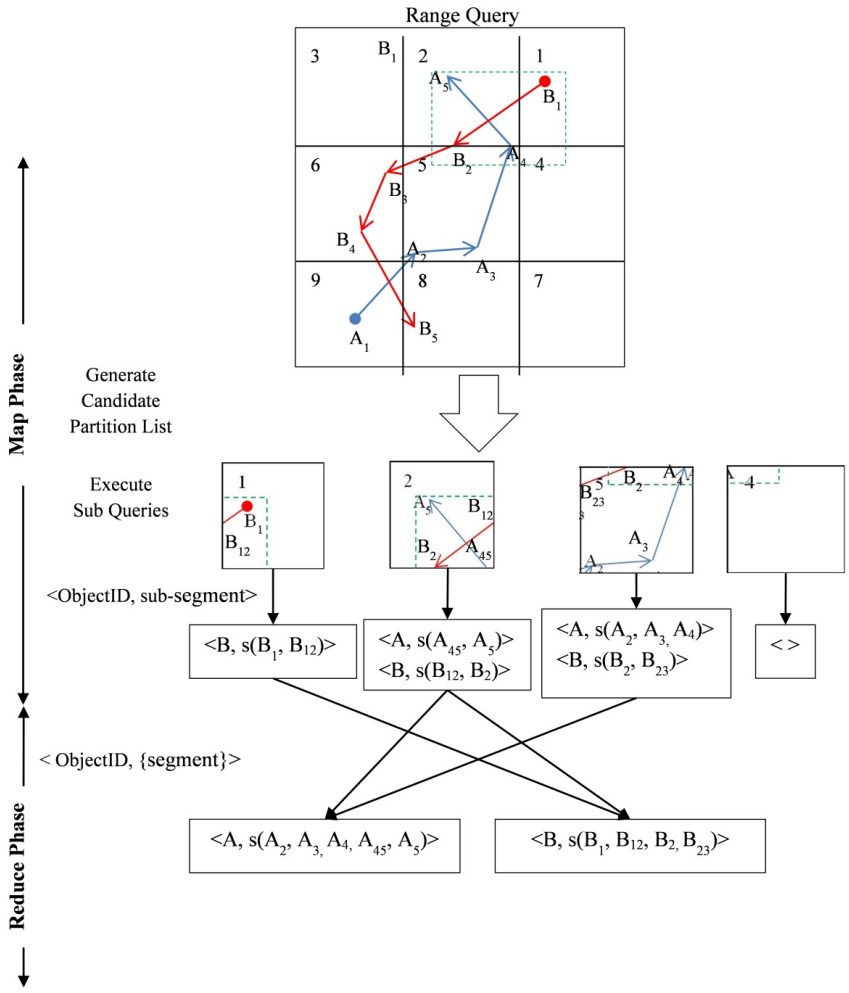


Figure 26: Spatiotemporal Range Query in PRADASE.

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

The S²T-Clustering sub-trajectory clustering method

As far as it concerns the sub-trajectory clustering mechanism, we propose S^2T Clustering, a four-step framework which consists of effective solutions to the problems of (i) trajectory voting and segmentation, (ii) sub-trajectory sampling, and (iii) sub-trajectory clustering and outlier detection. The motivation behind S^2T Clustering is that we seek for a methodology that segments and selects those sub-trajectories from a MOD that preserve as much as possible the properties and the mobility patterns hidden in the original MOD.

The problem of sub-trajectory clustering in a MOD is formalized according to the following definition:

Definition ***Sub-trajectory clustering in a MOD****. Assuming a dataset* $D = \{T_1, T_2, \ldots, T_N\}$ *consisting of N trajectories, each of which can be considered as a sequence of successive sub-trajectories each of arbitrary length* (i.e. $P_{k,i}$ *is the i-th sub-trajectory of trajectory* T_k)*, the optimal sub-trajectory clustering is the one that partitions the set of sub-trajectories into a clustering* $C = \{C_1, \ldots, C_M\}$ *of M clusters and a set Out of outliers. Assuming that each cluster* C_j *is represented by its representative (or centroid or typical) sub-trajectory* R_j, $j = 1, \ldots, M$*, the optimization criterion is to maximize the following expression, that gives the sum of representativeness of the entire dataset (SRD) using the set* $S = \{R_1, \ldots, R_M\}$ *of the representative sub-trajectories and the appropriate corresponding M clusters* $C(R_j)$*. The term* $\overline{V(P_{k,i}, R_j)}$ *implies the mean similarity* (*i.e. average number of votes, according to our terminology*) *that* $P_{k,i}$ *sub-trajectory has w.r.t. representative sub-trajectory* R_j:

$$SRD = \sum_{R_j \in S} \sum_{P_{k,i} \in C(R_j)} \overline{V(P_{k,i}, R_j)}$$

Note that maximization of the above expression is a hard problem as one first has to define the criterion with which a trajectory is partitioned to sub-trajectories, the technique for selecting the set of the most representative sub-trajectories whose cardinality *M* is unknown, to name but a few challenging sub-problems. In Figure 27 we outline our proposal that provides a solution to the above problem. The proposed S^2T-Clustering algorithm relies on a voting and segmentation process that detects homogenized sub-trajectories in the MOD, then selects the most representative ones to serve as the seeds of the clusters, and around them, forms the clusters (and isolates the outliers).

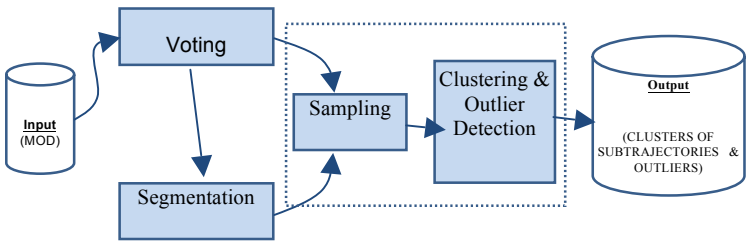


Figure 27: Scheme of the proposed sub-trajectory clustering framework.

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

Step 1 & 2: trajectory voting and segmentation

To perform voting-based trajectory segmentation, we adopt and appropriately adapt the Global Voting Algorithm (GVA) and the Trajectory Segmentation Algorithm (TSA), both proposed in [39]. The input of GVA is a MOD $D = \{T_1, T_2, \ldots, T_N\}$ indexed by a R-tree-like structure, a trajectory $T_k \in D$ and the $\sigma > 0$ parameter. The output of GVA is a vector V_k consisting of L_{k-1} components, which can be considered as a trajectory descriptor along the line segments $e_{k(i)}$, $i \in \{1,2,\ldots,L_{k-1}\}$, of trajectory T_k. As such, the components of vector $V_{k(i)}$ correspond to the voting (representativeness) received by the 3D line segments for each $e_{k(i)}$ of T_k.

The result of the voting process is the input to the trajectory segmentation algorithm TSA. The input of TSA is the normalized trajectory voting signal V_k, and two intrinsic parameters w, τ. The normalization is done by dividing V_k by the maximum of V_k, thus bounding $V_k \leq 1$. Parameter w sets the minimum size of a partitioned trajectory; as such it expresses the minimum number of line segments that can define a sub-trajectory. Parameter τ is related with the segmentation sensitivity of our method. As τ increases, the number of sub-trajectories reduces. It holds that τ can be set as a positive number close to zero (e.g. 0.01). The output of the method is the segmentation P_k of T_k into LP_k partitions, where LP_k is automatically estimated by the proposed scheme.

Step 3: sub-trajectory sampling

Having segmented the trajectories of a MOD according the above (voting-and-segmentation) process, we perform sampling in order to select the top-M representative sub-trajectories. For this purpose, the Sub-trajectory Sampling Algorithm (SSA), described in [39], is adopted. The input of SSA is the set of sub-trajectories P_k as segmented by TSA, the voting $VP_{k,i}$ and the normalized lifespan $NI_{k,i}$ vectors of these sub-trajectories. The output of SSA is the sub-trajectory sampling set S consisting of M samples. It is worth to note that M is not user-defined; in contrast, it is dynamically estimated by SSA. As such, SSA provides a deterministic solution.

Step 4: sub-trajectory clustering and outlier detection

SSA results are used to define the clusters. To this end, we propose the so-called Optimal Sub-trajectory Clustering Algorithm (SCA^O) that is presented in the following paragraphs. The algorithm is called Optimal since it is based on a sampling process that selects the best candidates via an optimization process.

Let S denote the sampling set, so that $S_{k,i}$ is one, if $P_{k,i}$ sub-trajectory belongs to the sampling set, and zero otherwise. The input of SCA^O are the segmented sub-trajectories $P_{k,i}$, $\forall\, k \in \{1, \ldots, N\}$, $i \in \{1, \ldots, LP_k\}$, the sampling set S of SSA, and the parameter ϵ. The output of the method is the clustering C and the outlier set Out of sub-trajectories.

First, each cluster is initialized by a sub-trajectory from the sampling set. Each sub-trajectory of the sampling set constitutes the first member (seed) of the corresponding cluster ($Cn(1)$, $n \in \{1, .., M\}$). Then, each sub-trajectory $P_{k,i}$ that does not belong to sampling set S is assigned to the cluster that

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

belongs, taking into account the mean value of the voting vector values $V(P_{k,i}, P_{m,n})$ and the parameter ϵ discussed earlier. In our experiments we have used $\epsilon = 0.1$. Lower values had insignificant changes in the clustering outcome.

The S^2T-Clustering algorithm so far solves the problem of sub-trajectory algorithm as an optimization problem. However, real-world applications may impose additional constraints, which are contradicting with the method used that optimally solves the problem. As in our case optimization takes place at the sampling step, in order to provide a constraint-aware algorithm we need to revisit the sampling process and prune out of the clustering step those candidates sub-trajectories (i.e. representatives around which clusters are formed), which cannot satisfy the initially posed constraints. To succeed this we unify in one step the sampling and clustering steps in one (as implied by the dotted line in Figure 27). More specifically, we adapt the SSA algorithm so as in each step to select the optimal representative sub-trajectory, however this representative in order to be added to the sampling set S, it should first satisfy the constraints of the group of sub-trajectories which is formulated according to the SCA^O specifications. Otherwise, this initially selected sub-trajectory is ignored and we search for the next candidate. Of course, this way we may scan the entire set of sub-trajectories without being able to identify enough clusters, meaning that we result in a huge set of outliers. In such a case we could relax the specifications for a sub-trajectory to belong to a cluster, by relaxing the parameter ε. Obviously, each scan of the remaining set of sub-trajectories re-examines sub-trajectories that have failed in a previous scan. This also implies that each scan results in a set of clusters belonging to an equivalence class of certain parameters. As such, the so-called Constraint-aware Sub-trajectory Clustering Algorithm (SCA^c) is able to control the size of the set of outliers, while at the same time satisfying any user-defined constraints for the resulted clusters. Of course, an implicit outcome of the algorithm is a different sampling set S, which now consists of the representatives around which can be formed clusters satisfying the placed constraints.

Furthermore, in [39] we showed that an efficient implementation of this demanding process requires the use of an incremental nearest neighbor (INN) algorithm, while the MOD is indexed by an R-tree-like structure. However, given the specifications we posed for the implementation of our MOD engine, this choice is not applicable due to that the access methods supported by real ORDBMS either do not allow the implementation of an INN algorithm, or their indexing extensibility interface has the discussed shortcomings. This implies that in our case we are forced to use classic NN queries (or range queries simulating NN ones by post-processing). In order to simulate the INN algorithm efficiently we propose the *Trajectory Buffer Query (TBQ)*. The TBQ takes as input a trajectory, calculates its trajectory buffer and returns the segments that overlap with it. Before defining the TBQ, it is necessary to define what a trajectory buffer is. The trajectory buffer in our setting, similarly to a trajectory, is a dynamic data type which, instead of segments, consists of the MBBs of the segments expanded by a spatial distance and a temporal interval. Formally, let as assume that T is a trajectory (composed of a sequence of segments), σ is a spatial distance and δ is a temporal interval. We define $TB(T, \sigma, \delta)$ to be a 3D 'buffer' around T with the property that every point in $TB(T, \sigma, \delta)$ is at most σ and δ (in space and time, resp.) far from a point in T. Hence, given a set S of trajectories, a reference

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

trajectory T, a spatial threshold σ and a temporal interval δ, the trajectory buffer query *TBQ(S, T)* retrieves those sub-trajectories in S having the property that the segments they are composed of, intersect with *TB(T, σ, δ)*. More implementation details about the TBQ are found in [7].

The Representative Trajectory Tree partitioning method

At this point, the connection between the sub-trajectory clustering method, proposed above, and the smart data partitioning technique, which will be the building block in our approach to tackle the Big data challenges posed within the project, is still fuzzy. In fact, the S^2T Clustering is a part of the smart data partitioning technique. Actually, the solution is given by a tree-like structure called *Representative Trajectory Tree (ReTraTree)*, which turns out to be able to (a) efficiently process both querying and mining operations in the MOD, (b) work in distributed mode, and (c) support trajectory databases that are fed in asynchronous and batch mode.

More specifically, we first partition each trajectory into $p << L_k$ equi-sized disjoint temporal periods (i.e. first level partitioning into so-called *chunks*), and secondly to organize each of the latter into possibly overlapping equivalence classes according to the lifespans of the sub-trajectories inside the chunks (i.e. second level partitioning into so-called *sub-chunks*).

The chunking process is applied incrementally whenever a batch of recordings from a moving object arrives. Then, the algorithm tries to fit it in the existing chunks, taking into consideration the already created chunking borders. If the given trajectory cannot be fitted in the existing temporal range, then the set of chunks is extended suitably in order to fit the trajectory.

At the second level, each chunk produced by the first phase is subdivided into (possibly) overlapping equivalence classes. Specifically, we partition each chunk into smaller sub-chunks by grouping the sub-trajectories contained in the chunk by their lifespan and their starting and ending timepoints. Therefore, in this phase the temporal borders of each sub-chunk, which are not defined by the user but from the data, are the same or similar w.r.t. a temporal tolerance parameter *tau* that is user-defined. This parameter implies that two sub-trajectories are considered temporally similar if their starting (ending) timepoints do not differ more than *tau*/2, respectively. Moreover, this parameter assumes that two sub-trajectories cannot be considered as spatio-temporally similar (as such they will not be in the same cluster and their distance function will not be calculated) if the union of their non-common lifespans is bigger than *tau* (i.e. a trajectory of 20 minutes duration cannot be similar with a trajectory of 30 minutes duration, when *tau*=10 minutes). Obviously, when setting *tau*=0 minutes, this will result into a large number of equivalence classes as in real-world data it is rare to have many trajectories starting their route absolutely concurrently. In Figure 28 the chunk corresponding to the first day (i.e. mauve colored sub-trajectories) is subdivided to two sub-chunks, containing $<T_1, T_2, T_3, T_4>$ and $<T_5, T_6>$ sets of sub-trajectories, respectively.

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

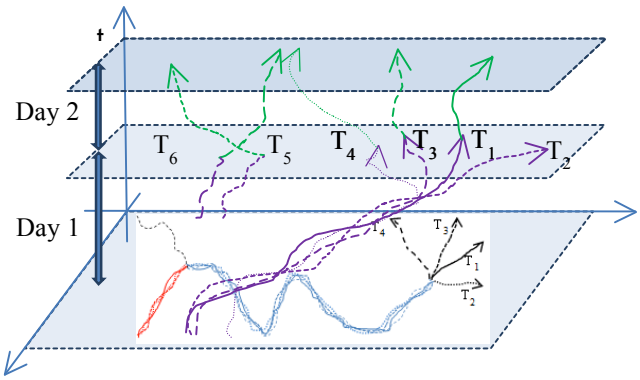


Figure 28: A MOD consisting of six trajectories

The ReTraTree data structure

The previous discussion regarding the hierarchical splitting of the time domain implicitly describes the first two levels of the ReTraTree structure (Figure 29). In detail, the root of the ReTraTree consists of entries corresponding to chunks sorted by time. Note that for each chunk CK_i there is no need to maintain the temporal periods in the index nodes as these correspond to equal-length splitting intervals. Each entry CK_i only maintains a pointer to the respective set of sub-chunks S_nCK_i, $n \geq 1$, forming the second level of ReTraTree. Each entry of a sub-chunk is a sequence of triplets $<S_nCK_i.per, S_nCK_i.S, S_nCK_i.Out>$, where *per* is the temporal period of the sub-chunk, while *S* (*Out*) are pointers to the set of representative (outlier) sub-trajectories of S_nCK_i. The sequence of triplets are ordered initially by the starting timepoint and secondly by the ending timepoint of *per*. The entries of the set *S* consist of pairs $<R_j, C_{R_j}>$, each of which include the representative sub-trajectory R_j and a pointer C_{R_j} to the subset of sub-trajectories that formulate a cluster around R_j. Similarly *S* is ordered by the time period of R_j. The set *Out* contains the outlier sub-trajectories of the current sub-chunk. The sets *S* and *Out* (whose utility and role will be discussed in the subsequent section) form a third level of partitioning in ReTraTree, while the actual data corresponding to all clusters C_{R_j} is the fourth level of the structure. Let's refer to this subset of data as $D_{n,i}$. Note that all the sub-trajectories of all C_{R_j} in a sub-chunk, namely $D_{n,i}$, are organized in a relation, whose column including the sub-trajectories is indexed by a 3DR-tree, while the column including the cluster identifier is indexed by a B+ tree. Of course, the relation further includes the identifiers of the trajectories, also indexed by a B+ tree. Obviously, these indices enable us to apply spatio-temporal queries to sub-chunk $D_{n,i}$ and facilitate the direct access into data of a specific cluster C_{R_j}.

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

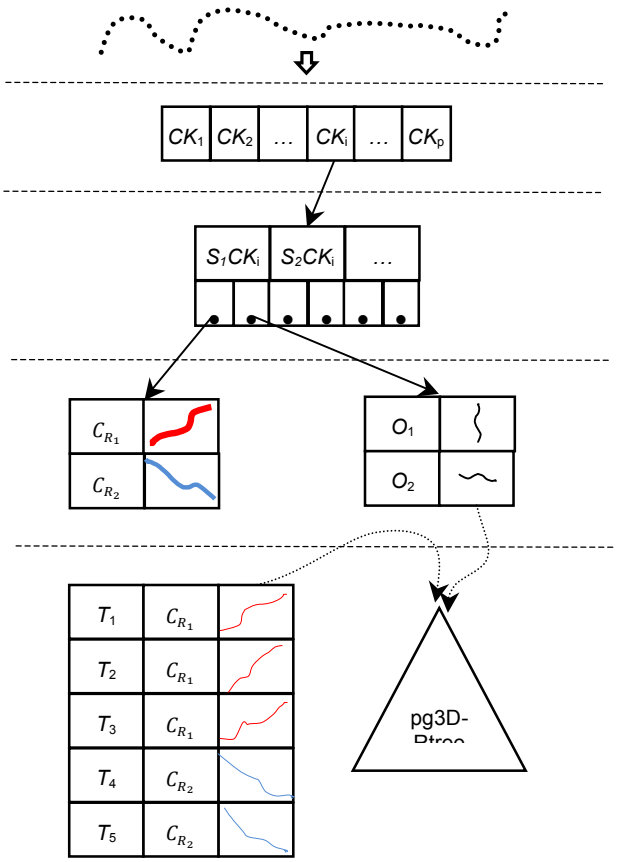


Figure 29: The structure of the ReTraTree

Incremental maintenance of ReTraTree

Our goal is to incrementally maintain the ReTraTree whenever a batch of recordings of a moving object (i.e. a trajectory T_k) arrives. Briefly, the rest of the methodology assigns each sub-trajectory to an appropriate sub-chunk. If there is not a matching sub-chunk w.r.t. time, a new subchunk is created, which is initialized with an empty representative set *S*, and an outliers set *Out* including the unmatched sub-trajectory. If there is an appropriate sub-chunk for the sub-trajectory under processing, the algorithm tries to assign it to an existing cluster. If this attempt fails, then the algorithm adds the sub-trajectory into the outliers' set, which act as a temporary relation upon which sampling-based sub-trajectory clustering (i.e. S^2T-Clustering Algorithm) is applied whenever the size of the relation exists a user-defined threshold (e.g. > α Mb). When this process takes place, a resulting new representative sub-trajectory will extend the existing set of representatives, only if it is different from them. Subsequently, for each of the resulting new outlier sub-trajectories, we either delete them (store them in a permanent outliers' relation) if their size is smaller than *w*, which means that it will not be able to be clustered in a future clustering round, or we re-drop the sub-trajectory from the top of the

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

ReTraTree structure. This implies that we recursively apply the procedure for that sub-trajectory (till it is either clustered or partitioned to smaller pieces, due to successive applications of the S^2T-Clustering algorithm) in order to search for other sub-chunks wherein the latter could be clustered or to form a new sub-chunk.

On the generality of ReTraTree

At this point, it would be good to investigate what kind of queries and mining operators could exist under such a structure. An obvious and general answer would be that time aware queries and operators would be answered efficiently. More specifically, we can have simple queries and operators such as the temporal range query and the time slice query, which are trivial. Furthermore, the spatiotemporal range query could be supported, which is also trivial but more expensive. A more advance operator could be the query by Trajectory-Example, where given a trajectory we can retrieve all the trajectories that "live" during the same period as the one given as input and are similar with it, based on the similarity function incorporated in the clustering algorithm employed by the structure. Another interesting operation that could be performed by the ReTraTree structure is, given a temporal period to retrieve the clusters (i.e. sub-trajectories "following" the representatives) that overlap this period. But this is not enough as it is possible that the sub-trajectory clustering process of overlapping sub-chunks to form clusters, namely representatives that: (a) are almost identical (as such, a *'merge'* process should take place in order to report only one cluster as the union of the two clusters built around the two similar representatives), and/or (b) one representative can be the continuation of another (as such, an *'append'* process should take place to identify maximal clusters). In other words what we require is a methodology that takes as input the ReTraTree structure as input and searches it, so as to identify maximal patterns w.r.t. the user requirements. Such a user requirement could be the discovery of all the valid clusters during a specific period of time (eventual this period could be the whole lifespan of the MOD, providing a solution also for the whole- (vs. sub-) trajectory clustering problem. This is a reasonable requirement in the BIG mobility data setting that we envision and the fact that state-of-art clustering algorithms are not able to be applied in the currently available MOD sizes. To put differently, the proposed methodology implies an algorithm that will act as a query operator in a MOD engine and that it will retrieve already clustered data according to user parameters and it will perform the afore-mentioned necessary merge and append refinements on the query results. To the best of our knowledge, such a query-based clustering approach is novel in the mobility data management and mining literature. More details about the QUery-based Trajectory clustering (QUT-clustering) are found in [15].

From ReTraTree to BIG data architectures

At this point, it's time to present the two approaches that are being explored in order exploit the above overviewed smart data partitioning technique. Up to now, ReTraTree was presented as an incremental clustering mechanism and a single server indexing structure but it can be more than that. Let us assume a distributed environment, consisting of several nodes, where each of them will host the data

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

and the pg3D-Rtree index contained in one (or several if size allows it) sub-chunk. Such a system, would achieve great scalability in answering the already mentioned time aware queries and mining operators.

Our tactic was not to build a system from scratch but to utilize existing systems by properly modifying them in order to fit our needs. In the first approach we utilize a database partitioning system implemented as procedural language, called PL/Proxy. In fact PL/Proxy is a new language created inside the PostgreSQL database that enables to make “sharding”/horizontal partitioning. The reason for using PL/Proxy is compatibility, since our Hermes MOD, is built upon PostgreSQL. Furthermore, by using PL/Proxy we can achieve storage of unstructured data, flexible distribution of data inside the cluster, simple handling and superior scalability while maintaining the transactional integrity provided by SQL. Each database (Master or Slave) will also have the Hermes extension installed instead of just PostgreSQL. The basic idea is to distribute the data in a way that each sub-chunk created by the ReTraTree structure along with the corresponding pg3D-Rtree index will be forwarded to a slave database, as shown in Figure 30.

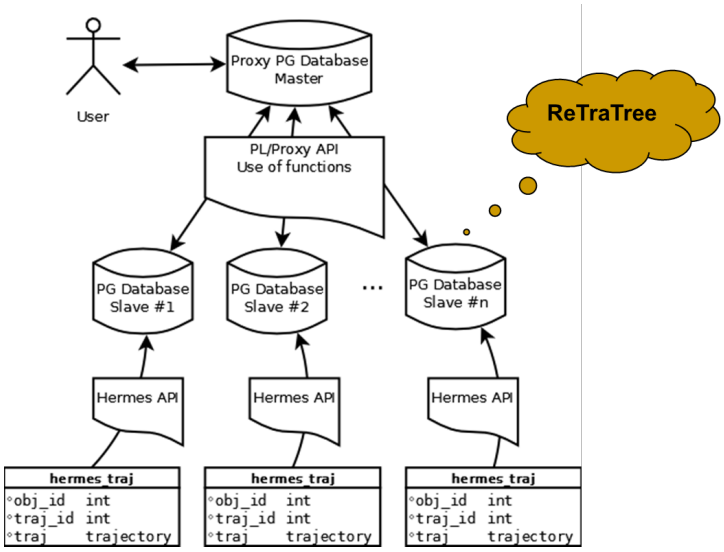


Figure 30: Towards a parallel MOD engine

The second approach explored is HadoopDB. As already described, HadoopDB tries to connect a set of single-node database systems by employing Hadoop as the network communication layer and task coordinator. Queries are basically parallelized across the nodes via the MapReduce framework; nevertheless, HadoopDB tries to push as much of the single node query work as possible inside the corresponding node databases. Our approach is to modify HadoopDB by installing in each node a PostgreSQL server instance along with the Hermes extension. Furthermore, our plan is to modify the Data Loader in such a way that it will incorporate the ReTraTree data partitioning logic. Finally, in order for the data types and function of Hermes to be “visible” to the user at the SQL level, wrappers should be written.

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

Similarly with the first approach, the basic idea is to distribute the data in a way that each sub-chunk created by the ReTraTree structure along with the corresponding pg3D-Rtree index will be forwarded to a node that includes a Hermes-enabled PostgreSQL database, as shown in Figure 31.

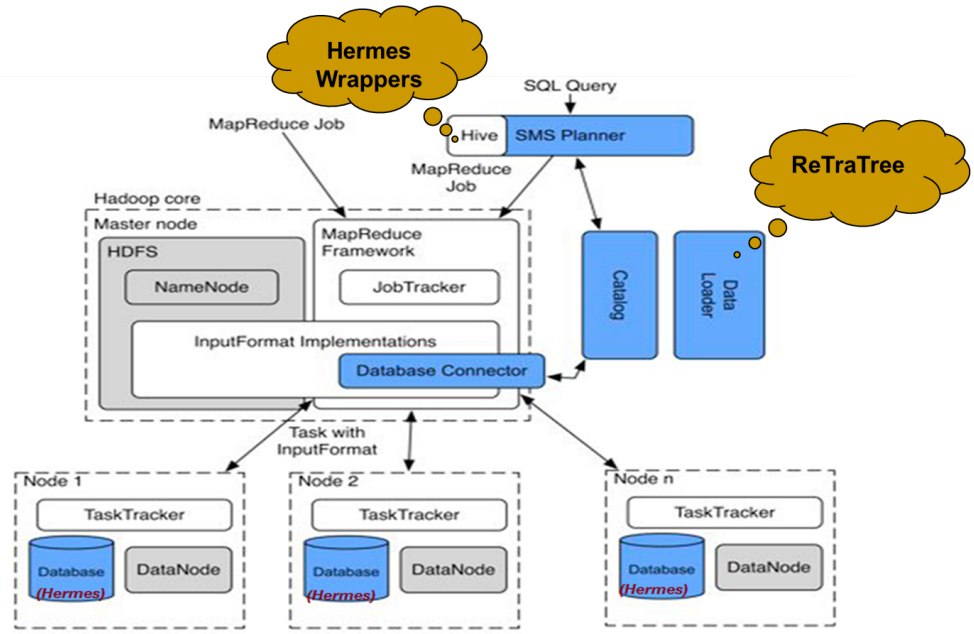


Figure 31: Hermes…with HadoopDB and Hive

These two approaches are already implemented and we are at the stage of experimenting with various tasks and queries.

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

6. Big Data Harmonization and Knowledge Integration

The manifold techniques to record mobility information have led to a variety of mobility data sets differing in their observation space, sampling coverage and resolution. In order to integrate and/or compare such data sets, these properties have to be harmonized. The easiest solution for harmonization is to summarize information for each dimension of movement data (spatial, temporal, population dimension, physical/semantic movement characteristics) of all more detailed data sets until they reach the level of the most general data set. However, this approach has the disadvantage that all specific information of the different data sets is lost. We therefore selected an opposite approach of *spatial disaggregation* from the field of Small Area Estimation (SAE). SAE has become an area of increasing scientific attention in recent years. It is used to estimate statistics in certain small areas of interest for which reliable values cannot be produced, for example, because the cost to conduct a survey is prohibitive.

Within the DATASIM project we apply SAE in order to assign mobility characteristics to the population in the administrative district of Cologne, Germany. In this region we have a rich set of mobility data sources available (e.g. GPS, CATI, traffic frequencies, prospectively Bluetooth). In part these data sources are integrated during SAE, in part they will serve to validate the outcome of SAE applying the validation framework of WP4. Note that the application of SAE does not compete to the DATASIM microsimulation model. It is applied to explore data integration techniques and to provide a population model for testing validation techniques on data sources unique to the Cologne region.

Various techniques have been developed to tackle SAE. A good overview can be found in [47]. We have decided to use methods from Combinatorial Optimization (CO) for approaching the spatial disaggregation problem. In fact, we have applied CO techniques to produce a synthetic population of Germany in a long running industry project with the Deutsche Post AG. The main purpose of the project was to disaggregate consumer behaviour for the purpose of direct mail campaigns. For DATASIM we apply the same methodological framework. However, research effort is invested in adapting the data fusion process to the available data sources, and especially to the design of linking variables for mobile behaviour.

Problem Description

Simply put, we have to find an answer to the question “who lives where”. We are provided with two datasets, one being a representative (mobility) sample of the population of Germany and the other one a set describing the households. For reasons of privacy, we cannot access the data on the household level. Instead, the data are aggregated to so-called cells. A cell in our case corresponds to a Navteq street segment and typically contains 5 up to 100 households, and thus is a small area. The household data are geo-referenced so we refer to these data also as spatial data. The problem of assigning questionnaires from the sample to the cells has much in common with the Statistical Matching Problem, sometimes also called Data Fusion (for more details see [24]).

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

The Statistical Matching Problem is defined as follows. Let *A* and *B* be two samples drawn from the same population. The approach in statistical matching then is to identify three sets of random variables. A set **X** of variables common to both sets, a set **Y** that exists in *A* only, and a set **Z** that exists in *B* only. The idea is to link records from *A* and *B* in such a way that the values of their common variables are similar if not identical. The outcome of this linkage process is a set *C* of records each having variables from **X**, **Y**, and **Z**. Some crucial conditions have to be met for this process to yield a reliable result.

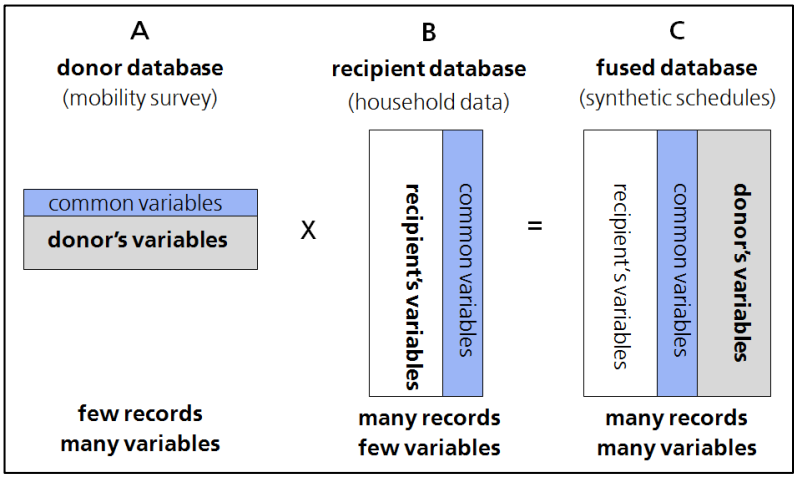


Figure 32: The statistical matching problem

In our case, however, the situation is different. The sets *A* and *B* are not drawn from the same population. The set *A* corresponds to the mobility sample of the population of Germany while the set *B* contains households (see Figure 32). More precisely, a record in *A* describes a person and a record in *B* describes a set of persons. So in the end we are facing a packing problem from the class of knapsack problems. These problems are known to be NP-hard. Heuristics like Tabu Search, Simulated Annealing or Evolutionary Algorithms are applied to solve them. Let n_c be the number of cells in the spatial data. We then have to solve n_c knapsack problems. To make things more complicated, these knapsack problems are not independent from each other. Since *A* is a representative sample one would like the set of all questionnaires assigned to cells to be representative, too. Technically speaking, we want to preserve the marginal distribution (PMD). If a questionnaire q stands for n_q people, q should appear n_q times in the resulting table *C*. The fitness function contains a term to assess the quality of the PMD. Note that without the PMD constraint one could solve the knapsack problems one after the other. With the PMD constraint present the knapsack problems are loosely coupled. Thus, they are dealt with by the solver all at the same time.

We decided to design and implement a Genetic Algorithm. Modern day multi-core computers perfectly support parallel computing so it was clear from the beginning that we will realise a parallel GA. Various papers (cf. [49][32]) suggest that the CO approach is advantageous compared with the statistical methods. This confirmed our belief a PGA is the right way to tackle this problem.

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

Situation in DATASIM

For the use in DATASIM we have a sample survey of 2.659 persons in the administrative district of Cologne. The data set is provided by the Arbeitsgemeinschaft Media-Analyse e.V. (Ag.ma), a joint industry committee of German advertising vendors and customers, and has been described in more detail in D1.1.

Spatial information is taken from various statistics as well as spatial data sets describing the households and their spatial neighborhood. In total, the administrative district of Cologne contains 314.305 Navteq street segments with a population of about 3.4 million inhabitants (above 14 years of age).

A variable from the set **Y** present only in the survey is the mobility pattern. A variable from the set **Z** that exists in the spatial data only is the geo-coordinate. When linking the survey with the household data each household will be assigned a set of mobility patterns, one for each person living in the household.

We are currently adapting the existing data fusion process and GA to the needs in DataSim. This means primarily to identify variables that can be used to link both data sets. In addition, some changes to the fitness function are required.

The workflow is as follows:

- *Identification of the common variables*: Good socio-demographic candidates are gender, age, income, employment status and car ownership. In addition, regional variables as urban or rural location or proximity to shopping facilities are correlated to travel behavior [22].
- *Clustering of the spatial data*: For technical reasons, the optimizer expects the spatial data to be clustered according to the two or three most significant common variables.
- *Preprocessing the data*: Some simple statistical values need to be computed for use in the fitness function. The values of some of the common variables are transformed into classes.
- *Starting the optimizer*: The parallel GA is started with 64 processes and 16 individuals on each process. Local search is done by an Iterated Local HillClimber which focuses on the PMD.

Beside the above-described core activity regarding the data harmonization and integration task (i.e. Task 1.3), we have pursued two more activities that fit also to the Task1.2 but also serve the objectives of other WPs. These activities resulted into the Hermoupolis semantic trajectory synthetic generator [8] and the EasyTracker [1] semantic trajectory mobile annotation tool.

Hermoupolis: A Trajectory Generator for Simulating Generalized Mobility Patterns

Although we have in our possession a unique dataset, which combines activity diaries and GPS data in a synchronized way (see Section 2.2 at D1.1), this dataset is small for performing efficiency experiments. What is more, such a synchronized, dual dataset provides ground truth for the properties

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

and the knowledge hidden inside each of two different facets. Having GPS data, and knowing the home addresses of individuals, we may validate the activities took place at home, while having activity diaries, we may validate algorithms that perform the STOP (MOVE) discovery and at a second step semantic annotation of the discovered STOP (MOVE). To tackle this issue, we designed an appropriate generator of synthetic semantically-enriched mobility datasets, called Hermoupolis [8].

Briefly, Hermoupolis exploits on well-known synthetic trajectory generators by first extracting the properties and the patterns hidden inside a small real world dataset (as the one we have in our possession) and then by using these generators so as to simulate and extrapolate the discovered properties at larger scales. This process is highlighted in Figure 33. Hermoupolis takes as input: (a) a road network (whereupon the generated objects will move), (b) an application depended geographically-referenced land use database and (c) the derived mobility profiles extracted by the previously discussed knowledge discovery process. The output of the generator is such synchronized datasets of user-defined volumes.

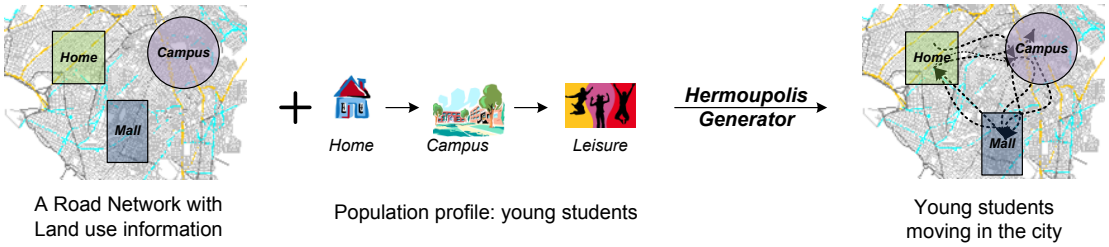


Figure 33: An example of simulating young students moving in the city, generated by Hermoupolis.

EasyTracker: An Android application for capturing mobility behavior

In order to be able to have such synchronized datasets, but real ones, we have developed "EasyTracker" [1], an Android mobile application for capturing mobility behavior. The core of EasyTracker is threefold. At first, EasyTracker collects space-time points of user's movement using a GPS receiver, thus creating and visualizing a path (or trajectory) followed by the user. In contrast to similar applications, the collected path, to be stored in a server's database, is sanitized by filtering out locations that lie inside user-defined areas of interest – sensitive areas where the user does not allow to be tracked (e.g. around home, a hospital, etc.). This way, EasyTracker provides end users with personalized privacy functionality.

The second key functionality is that EasyTracker allows a user to annotate parts of her track with labels and therefore to describe her current activity (e.g. "stopped at café A", "driving towards office"). Such feature can turn out to be very useful for automatic fill-in of surveys performed in transportation science or for researchers working on activity recognition who are usually restricted to use manually processed surveys.

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

Third, EasyTracker encapsulates state-of-the-art algorithms that process in an online fashion the received stream of GPS recordings and transforms it into meaningful tracks, ready to be stored into a MOD for further analysis. Specifically, EasyTracker includes line simplification methods that compress the incoming stream of timestamped locations (thus reducing the storage cost), which are then partitioned into homogeneous portions according to some spatio-temporal criteria using a state-of-the-art segmentation method [39]. According to this segmentation, the track is split into portions (i.e. sub-tracks), which can be labeled by tags that describe the corresponding spatio-temporal behavior of the user (e.g. STOPPED, when the speed is very low). This is important as it facilitates the user (or some auditing algorithm) to compare her manual annotations with the classified sub-tracks as provided by the segmentation algorithm. The big picture that illustrates these novel features of EasyTracker is depicted in Figure 34.

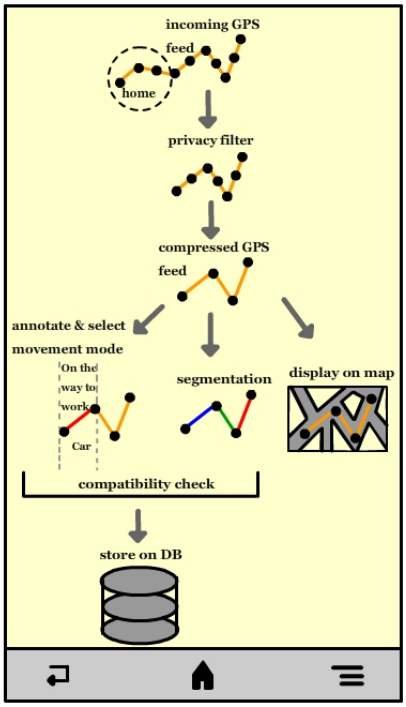


Figure 34: The big picture of EasyTracker

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

7. REFERENCES

Category 1: published papers that acknowledge Datasim EU funding.

[1] A. Doulamis, N. Pelekis and Y. Theodoridis. "EasyTracker: An Android application for capturing mobility behavior", Proceedings of the 2nd workshop on Mobile Device Software Development and Web Development (MDSD), organized in conjunction with the 16th Panhellenic Conference on Informatics (PCI 2012), Piraeus, Greece, 2012.

[2] D. Kopanaki, N. Pelekis, A. Gkoulalas-Divanis, M. Vodas and Y. Theodoridis. "A Framework for Mobility Pattern Mining and Privacy-Aware Querying of Trajectory Data", Proceedings of the 11th Hellenic Data Management Symposium (HDMS'12), Chania, Greece, 2012.

[3] N. Pelekis, A. Gkoulalas-Divanis, M. Vodas, A. Plemenos, D. Kopanaki and Y. Theodoridis. "Private-HERMES: A Benchmark Framework for Privacy-Preserving Mobility Data Querying and Mining Methods", Proceedings of the 15th International Conference on Extending Database Technology (EDBT'12), Berlin, German, 2012.

[4] G. Kellaris, N. Pelekis, Y. Theodoridis: "Map-Matched Trajectory Compression", Journal of Systems and Software, 86(6):1566-1579, June 2013. Elsevier.

[5] R. Fileto, M. Krüger, N. Pelekis, Y. Theodoridis, C. Renso: "Baquara: A Holistic Ontological Framework for Movement Analysis with Linked Data", Proceedings of the 32nd International Conference on Conceptual Modeling, ER'13, Hong Kong, November 2013. Springer.

[6] E. Frentzos, Nikos Pelekis, Nikos Giatrakos, Yannis Theodoridis: "Cost Models for Nearest Neighbor Query Processing over Existentially Uncertain Spatial Data", Proceedings of the 13th Int'l Symposium on Spatial and Temporal Databases, SSTD'13, Munich, Germany, August 2013. Springer.

[7] B. Krogh, O. Andersen, E. Lewis-Kelham, N. Pelekis, Y. Theodoridis, K. Torp. "Trajectory Based Traffic Analysis (demo paper)", Proceedings of ACM SIGSPATIAL GIS Conference.

[8] N. Pelekis, C. Ntrigkogias, P. Tampakis, S. Sideridis, Y. Theodoridis: "Hermoupolis: A Trajectory Generator for Simulating Generalized Mobility Patterns", demo paper, Proceedings of the European Conference on Machine Learning / Principles and Practice of Knowledge Discovery in Databases, ECML/PKDD'13, Prague, Czech Republic, September 2013. Springer.

[9] N. Pelekis and Y. Theodoridis, "Mobility Data Management and Exploration", Springer, New York, 2013, in press.

[10] R. Wagner, A. Raffaeta, A. Roncato, J. A. de Macedo, R. Trasarti, and C. Renso. Mob-Warehouse: a semantic approach for mobility analysis with a trajectory data warehouse. SECOGIS workshop at ER conference, HongKong, 2013.

Category 2: submitted papers

[11] B. Krogh, N. Pelekis, Y. Theodoridis, K. Torp. "Strict Path Query Evaluation on Large Trajectory Databases", submitted to IEEE Int. Conference on Data Engineering (ICDE).

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

[12] N. Pelekis, D. Janssens, Y. Theodoridis. “When Bob Met Alice: from Trajectories to Semantic Mobility Networks”, submitted to ACM SIGMOD Record, under review.

Category 3: technical reports / working papers (tentative titles)

[13] N. Pelekis, S. Sideridis, Y. Theodoridis. “From Trajectories to Semantic Mobility Networks - Hands-On SBO survey dataset”. UNIPI-InfoLab-TR-2013-01, Technical Report Series, U. Piraeus, 2013. Available at :
http://infolab.cs.unipi.gr/technical_reports/TR-2013-01.pdf

[14] N. Pelekis, C. Panagiotakis, M. Vodas, P. Tampakis, Y. Theodoridis. “Sampling-based sub-trajectory clustering”. UNIPI-InfoLab-TR-2013-02, Technical Report Series, U. Piraeus, 2013. Available at :
http://infolab.cs.unipi.gr/technical_reports/TR-2013-02.pdf

[15] N. Pelekis, P. Tampakis, M. Vodas, Y. Theodoridis. “Incremental Sub-Trajectory Clustering of Large Moving Object Databases”. UNIPI-InfoLab-TR-2013-03, Technical Report Series, U. Piraeus, 2013. Available at :
http://infolab.cs.unipi.gr/technical_reports/TR-2013-03.pdf

[16] M. Vodas, N. Pelekis, Y. Theodoridis, C. Ray. “On Benchmarking a Trajectory DBMS”. UNIPI-InfoLab-TR-2013-04, Technical Report Series, U. Piraeus, 2013. Available at :
http://infolab.cs.unipi.gr/technical_reports/TR-2013-04.pdf

[17] D. Kopanaki, N. Pelekis, S. Sideridis, M. Vodas, A. Gkoulalas-Divanis, Y. Theodoridis. “Privacy-Aware Querying in Semantically-Enriched Trajectory Databases”. UNIPI-InfoLab-TR-2013-05. Available at :
http://infolab.cs.unipi.gr/technical_reports/TR-2013-05.pdf

Category 4: papers not acknowledging DATASIM, though during the DATASIM period

[18] G. Marketos, M. L.. Damiani, N. Pelekis, Y. Theodoridis, Z. Yan. “Trajectory Collection and Reconstruction”, In C. Renso, S. Spaccapietra, and E. Zimányi (editors), Mobility Data: Modeling, Management, and Understanding. 2013.

Category 5: other papers

[19] A. Abouzeid, K. Bajda-Pawlikowski, D. Abadi, A. Silberschatz, A. Rasin. HadoopDB: an architectural hybrid of MapReduce and DBMS technologies for analytical workloads. In Proceedings of VLDB Endowment, 2(1), 922-933, 2009.

[20] V. Bogorny, B. Kuijpers and L.O. Alvares. ST-DMQL: a semantic trajectory data mining query language. Int. Journal of Geographical Information Science, 23:1245-1276, 2009.

[21] V. Bogorny, C. Renso, A.R. de Aquino, F. de Lucca Siqueira and L.O. Alvares. CONSTAnT – A Conceptual Data Model for Semantic Trajectories of Moving Objects. Transactions of GIS, 2013.

[22] C. Curtis, T. Perkins. Travel Behavior: A review of recent literature. 2006.

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

[23] J. Dean, S. Ghemawat. MapReduce: simplified data processing on large clusters. In Proceedings of OSDI, 2004.

[24] M. D'Oracio, M.D. Zio, M. Scanu. Statistical Matching – Theory and Practice, 2006.

[25] F. Giannotti, and D. Pedreschi. Mobility, Data Mining and Privacy: Geographic Knowledge Discovery. Springer, 2008.

[26] F. Giannotti, M. Nanni, D. Pedreschi, F. Pinelli, C. Renso, S. Rinzivillo, R. Trasarti. Unveiling the complexity of human mobility by querying and mining massive trajectory data, VLDB J., 20(5): 695-719, 2011.

[27] Gkoulalas-Divanis, A. and Verykios, V. S. (2008). A privacy–aware trajectory tracking query engine. SIGKDD Explorations, 10(1):40-49.

[28] J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venkatrao, F. Pellow and H. Pirahesh. Data cube: A relational aggregation operator generalizing groub-by, cross-tab and sub-totals. DMKD, 1(1):29-54, 1997.

[29] R. H. Güting and M. Schneider. Moving Objects Databases. Morgan Kaufmann, 2005.

[30] R. H. Güting, T. Behr, C. Düntgen. SECONDO: A Platform for Moving Objects Database Research and for Publishing and Integrating Research Implementation. IEEE Data Engineering Bulletin, 33(2):56-63, 2010.

[31] J. Han, N. Stefanovic and K. Koperski. Selective materialization: An efficient method for spatial data cube construction, Research and Development in Knowledge Discovery and Data Mining, LNCS, Volume 1394, 144-158, 1998.

[32] Harland, Heppenstall, Smith, Birkin. Creating Realistic Synthetic Populations at Varying Spatial Scales: A Comparative Critique of Population Synthesis Techniques, 2012.

[33] D. Janssens, F. Giannotti, M. Nanni, D. Pedreschi, S. Rinzivillo. Data Science for Simulating the Era of Electric Vehicles. Künstliche Intelligenz, 26(3):275-278, 2012.

[34] J. Jiang, H. Bao, E.Y. Chang, Y. Li. MOIST: a scalable and parallel moving object indexer with school tracking. In Proceedings of VLDB Endowment, 5(12), pp. 1838-1849, 2012.

[35] Lee, J.-G., Han, J., and Whang, K.-Y. Trajectory clustering: a partition-and-group framework. In Proceedings of SIGMOD, 2007.

[36] Q. Ma, B. Yang, W. Qian, A. Zhou. Query processing of massive trajectory data based on mapreduce. In Proceedings of Cloud data management workshop, 2009.

[37] Nanni, M., and Pedreschi, D. Time-focused clustering of trajectories of moving objects. Journal of Intelligent Information Systems, 27(3):267 – 289, 2006.

[38] G. Marketos, E. Frentzos, I. Ntoutsi, N. Pelekis, A. Raffaetà, Y. Theodoridis. Building real-world trajectory warehouses. In MobiDE 2008, Seventh International ACM Workshop on Data Engineering for Wireless and Mobile Access, pages 8-15, ACM Press 2008.

[39] Panagiotakis, N. Pelekis, I. Kopanakis, E. Ramasso, Y. Theodoridis, "Segmentation and Sampling of Moving Object Trajectories based on Representativeness", IEEE Transactions on Knowledge and Data Engineering, 2011.

[40] L. I. Panis, S. Broekx, R. Liu. Modelling instantaneous traffic emission and the influence of traffic speed limits. Science of the Total Environment, 371, 270–285, 2006.

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

[41] Parent, S. Spaccapietra, C. Renso, G. Andrienko, N. Andrienko, V. Bogorny, M. L. Damiani, A. Gkoulalas-Divanis, J. Macedo, N. Pelekis, Y. Theodoridis, Z. Yan, “Semantic Trajectories Modeling and Analysis" ACM Computing Surveys, 45(4), December 2013. ACM Press.

[42] N. Pelekis, E. Frentzos, N. Giatrakos, and Y. Theodoridis. HERMES: Aggregative LBS via a trajectory DB engine. In Proceedings of SIGMOD, pages 1255-1258, 2008.

[43] N. Pelekis, A. Gkoulalas-Divanis, M. Vodas, D. Kopanaki, and Y. Theodoridis. Privacy-Aware Querying over Sensitive Trajectory Data. In Proceedings of CIKM, pages 895-904, 2011.

[44] Pfoser, C. S. Jensen. “Capturing the Uncertainty of Moving-Object Representations”. Proceedings of SSD, 111-132, 1999.

[45] D. Pfoser, C.S. Jensen, Y. Theodoridis. Novel Approaches to the Indexing of Moving Object Trajectories. In Proceedings of VLDB, 2000.

[46] I. S. Popa, K. Zeitouni, V. Oria, D. Barth, and S. Vial. Indexing in-network trajectory flows. VLDB J., 20(5):643–669, 2011.

[47] A. Rahman. A Review of Small Area Estimation Problems and Methodological Developments. Discussion Paper, University of Canberra, 2008.

[48] C. Renso, M. Baglioni, J.A.F. de Macedo, R. Trasarti, M. Wachowicz. How you move reveals who you are: understanding human behavior by analyzing trajectory data. Knowledge and Information System Journal (KAIS), pp. 1-32, 2012.

[49] Ryan, Maoh, Kanaroglou. Population Synthesis: Comparing the Major Techniques Using a Small, Complete Population of Firms, 2007.

[50] S. Spaccapietra, C. Parent, M.L. Damiani, J.A. Macedo, F. Porto and C. Vangenot. A conceptual view on trajectories. Data & Knowledge Engineering 65: 126–146, 2008.

[51] S. Spaccapietra and C. Parent. Adding meaning to your steps, In Proceedings of ER, 2010.

[52] L. Spinsanti, F. Celli and C. Renso. Where you stop is who you are: understanding peoples' activities. In Proceedings of BMI, User Behaviour Modelling, 2010.

[53] H. Tan, W. Luo, L.M. Ni. CloST: a hadoop-based storage system for big spatio-temporal data analytics. In Proceedings of CIKM, 2012.

[54] M. Taylor, J. Woolley, and R. Zito. Integration of the global positioning system and geographical information systems for traffic congestion studies. Transportation Research Part C: Emerging Technologies, 8(1-6):257–285, 2000.

[55] O. Wolfson, H. Cao, H. Lin, G. Trajcevski, F. Zhang, N. Rishe. Management of Dynamic Location Information in DOMINO. In Proceedings of EDBT, 2002.

[56] D. Wu, G. Cong, C.S. Jensen. A framework for efficient spatial web object retrieval. The VLDB Journal, 21:797–822, 2012.

[57] Z. Yan, J.A. Macedo, C. Parent, S. Spaccapietra. Trajectory Ontologies and Queries. Transactions in GIS, 12(Suppl. 1): 75–91, 2008.

[58] Z. Yan, D. Chakraborty, C. Parent, S. Spaccapietra and K. Aberer. SeMiTri: A Framework for Semantic Annotation of Heterogeneous Trajectories. In Proceedings of EDBT, 2011.

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833

[59] Z. Yan, C. Parent, S. Spaccapietra and D. Chakraborty. A Hybrid Model and Computing Platform for Spatio-Semantic Trajectories. In Proceedings of ESWC, 2010.

[60] Z. Yan, N. Giatrakos, V. Katsikaros, N. Pelekis and Y. Theodoridis, Y. SeTraStream: Semantic-aware Trajectory Construction over Streaming Movement Data. In Proceedings of SSTD, 2011.

[61] P. Zhao, X. Li, D. Xin and J. Han. Graph Cube: On Warehousing and OLAP Multidimensional Networks. In Proceedings of ICDE, 2012.

[62] Y. Zheng, Y. Chen, X. Xie and W.-Y. Ma. Understanding transportation modes based on GPS data for Web applications, ACM Transactions on the Web, 4(1), 2010.

Project funded by the European Community under the Information and Communication Technologies Programme - Contract ICT-FP7-270833