# Dependency Rule Modeling for Multiple Aspects Trajectories

Ronaldo dos Santos Mello[1][0000−0003−4262−5474], Geomar André Schreiner[1][0000−0002−6457−8982], Cristian Alexandre Alchini[2], Gustavo Gonçalves dos Santos[2], Vania Bogorny[1][0000−0002−0159−4643], and Chiara Renso[3][0000−002−1763−2966]

[1] UFSC, PPGCC, Florianópolis, Brazil {r.mello,vania.bogorny}@ufsc.br schreiner.geomar@posgrad.ufsc.br
[2] UFSC, INE, Florianópolis, Brazil {cristian.alchini,gustavo.gs} @grad.ufsc.br
[3] ISTI-CNR, Pisa, Italy
chiara.renso@isti.cnr.it

**Abstract.** Trajectories of moving objects are usually modeled as sequences of space-time points or, in case of semantic trajectories, as labelled stops and moves. Data analytics methods on these kinds of trajectories tend to discover geometrical and temporal patterns, or simple semantic patterns based on the labels of stops and moves. A recent extension of semantic trajectories is called *multiple aspects trajectory*, *i.e.*, a trajectory associated to different semantic dimensions called *aspects*. This kind of trajectory increases in a large scale the number of discovered patterns. This paper introduces the concept of *dependency rule* to represent patterns discovered from the analysis of trajectories with multiple aspects. They include patterns related to a trajectory, trajectory points, or the moving object. These rules are conceptually represented as an extension of a conceptual model for multiple aspects trajectories. A case study shows that our proposal is relevant as it represents the discovered rules with a concise but expressive conceptual model. Additionally, a performance evaluation shows the feasibility of our conceptual model designed over relational-based database management technologies.

**Keywords:** Multiple aspect trajectory · conceptual model · data analytics · dependency rule.

## 1 Introduction

Mobility data modeling is receiving more and more attention in the recent years due to the increasing easiness to collect data from mobile applications. In the beginning of the 2000 decade, trajectories of moving objects were modeled as sequences of points with space and time information (the so-called *raw trajectories*) [9]. From 2007, a new view over trajectory data called *semantic trajectory* was proposed. It is represented not only in terms of space and time dimensions, but also *stops* and *moves*, to denote parts of trajectories where the object stayed for a certain amount of time or changed its position, respectively [15].

More recently, new data models were conceived to represent semantic trajectories not only as stops and moves, but with other predefined semantic dimensions, like the goal of the trip, the purpose of a visit, or the transportation mode [5,8]. This concept further evolved to the general notion of *multiple aspects trajectory (MAT)*, where the semantic dimensions are not predefined, *i.e.*, it is possible to associate any kind of enrichment information to a trajectory. The pioneer work of Mello et al. [11] introduced a conceptual model for MATs, called *MASTER*, where the semantic dimensions are called *aspects*.

As an example of a MAT, imagine the movement of a person during a weekend day. He/she leaves home, goes to a park and then to a restaurant. The person has a smart watch that constantly collects *blood pressure rate* and *body temperature*. The park, in turn, have *open and close hours*. The weather condition may change during his/her movement (*e.g.*, from *sunny* to *rainy*), as well as the used transportation modes (*e.g.*, *train* and *taxi*). This example highlights how several heterogeneous aspects may coexist in a MAT.

On going to the analysis of MAT data, we may see several challenges related to knowledge discovery as the behaviour of a moving object may involve several aspects and, additionally, some aspects may be strongly correlated (or dependent) and may not be analyzed separately. For instance, suppose the restaurant visited by the person aforementioned has a *spatial location*, a *category*, some *reviews*, *average price*, and *rating*. These last three attributes may hold a *dependency* stating that ratings equal to *10* have average price higher than *U$ 100* and *excellent* or *good* reviews. We call these dependency relationships between attributes as a *dependency rule (DR)*. A DR may be learned through data mining or machine learning methods, or it may be predefined by the user [12].

Finding, representing and storing these dependencies is therefore an essential step when analysing MATs. Although there are advances in trajectory data modeling and mining, there is no consensus among approaches for modeling discovered patterns from trajectory data, and the existing ones have limitations. One example is the work of Bogorny et. al. [4], which models patterns for trajectories only in terms of stops and moves, and considers a few/fixed aspects.

This paper conceptually define *DRs* over MATs. For doing that, we extend the MASTER model [11]. Our main contributions are:

– we define the concept of DR as a pattern related to MAT data;
– we introduce a notation for expressing DRs with a power expression higher than the traditional association rules [1];
– we propose an extension of MASTER called *MASTER DR*. With MASTER DR it is possible to query the entities on which the DR holds or vice-versa;
– we provide an evaluation of our model that comprises a case study over real MAT data, as well as a performance experiment over relational-based Database (DB) technologies.

The rest of this paper is organized as follows. Section 2 provides a background about MAT as well as the MASTER model. Section 3 presents the related work and Section 4 introduces the concept of DR and its representation. Section 5 describes MASTER DR, Section 6 presents some evaluations and Section 7 is dedicated to the conclusion.

## 2 Multiple Aspect Trajectory and the MASTER Model

A *MAT* is a trajectory that may be enriched with an unlimited number of semantic information called *aspects* [11]. An *aspect* is a real-world fact that is relevant for trajectory data analysis, and it is characterized by an *aspect type*. For instance, the aspect *subway* belongs to an aspect type *transportation*, and an aspect *rainy* belongs to an aspect type *weather*. An aspect type act as a metadata definition for an aspect. It holds a set of *attributes* and it may also be a subtype of a more general aspect type, allowing an aspect type *subtypeOf* hierarchy, like *POI←accommodation←hotel*. We consider *time* and *space* as possible aspects. So, an aspect can be specialized into *Spatial Aspect* or *Temporal Aspect*. The first one holds *position* attributes (x,y), and the second one a *timestamp* attribute.

A MAT, in turn, is represented by a set of *points* that denotes the movement of a *moving object*, *i.e.*, a real-world entity that moves along space and time. This object is always associated to a *type*, which can be a person, a drone, an animal, a car, or even a natural phenomenon, like a hurricane.

The MASTER conceptual model combine simplicity and expressive power for representing aspects. The intention is to represent any semantic dimension, independent of the application domain. Figure 1 shows the last version of the MASTER model (the yellow entities inside the MASTER package).

An aspect is associated to a point when it changes frequently during the object movement. One example is a *visited place (a POI)*. When an aspect does not vary during an entire MAT, it is associated to the MAT as a whole. An example could be the *weather* condition. When an aspect holds during the entire life or a long period of a moving object, it is associated to the moving object.

Finally, MASTER introduces the *moving object relationship*. A moving object may maintain any type of relationship with other moving objects, and these relationships may also be characterized by different aspects such as the type of relationship (*e.g.*, *friendship*, *professional*, *family*).

## 3 Related Work

Previous approaches in the literature introduced the notion of semantically enriched trajectories, as the pioneer work of Spaccapietra et al. [15], as well as the *CONSTANT* [5] and *MASTER* [11] data models. It is worth also mentioning works that represent semantic trajectories and associated patterns. Some of them base their novelty in exploiting ontologies to represent both (semantic) trajectories and patterns [8, 14]. However, none of them consider dependencies among semantic data and how to represent them.

Some other works propose conceptual models for data mining patterns [2, 4, 7, 13, 15, 16]. On compared to our proposal, these works do not necessarily focus on trajectories, or represent trajectories as only spatio-temporal points or sequences of stops and moves, not including complex aspects, or considering a single semantic point of view. Due to it, the modeled patterns are limited.

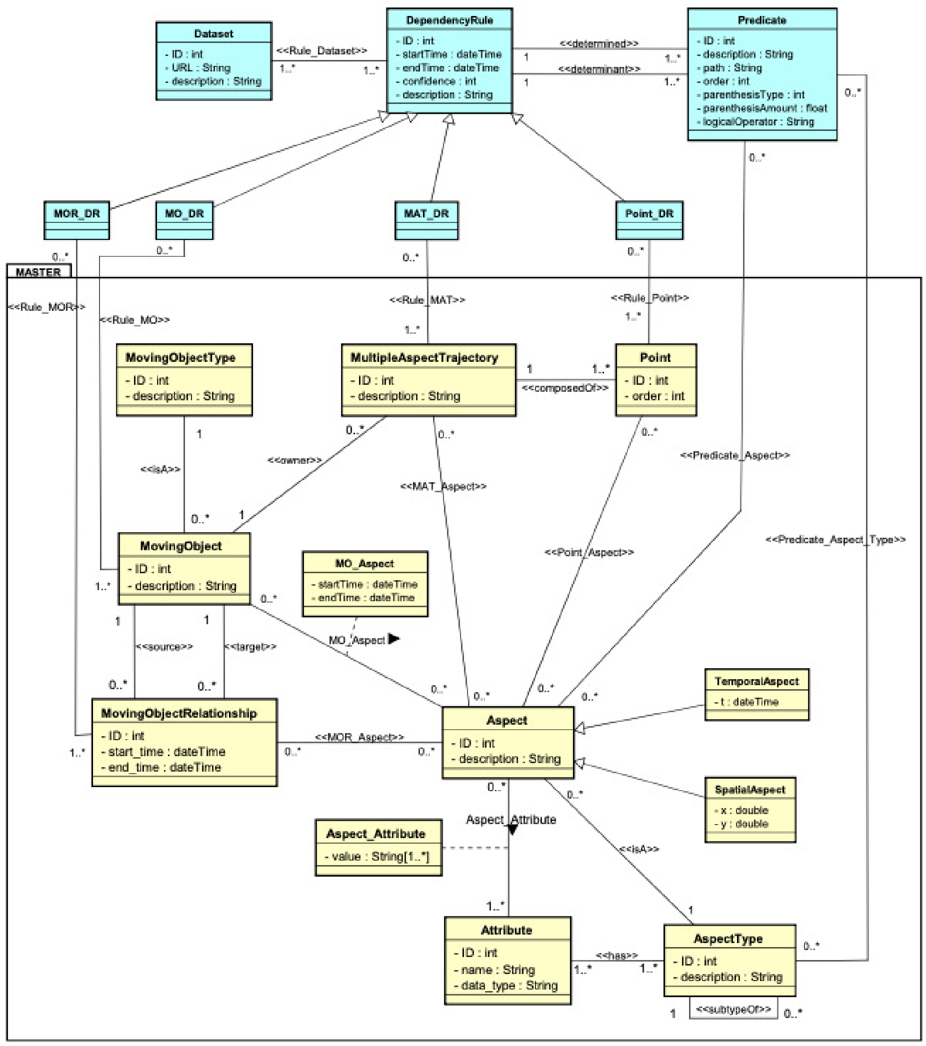


**Fig. 1.** The MASTER conceptual model and the dependency rule extension

This paper introduces DR as a pattern that is able to represent complex dependencies among trajectory aspects. DRs for MATs has not been considered in the literature so far as MATs is a relatively new trajectory concept, and in previous definitions of semantic trajectories the enrichment component is just a label, being not suitable to discovery dependencies among aspects.

## 4 Dependency Rule

A *DR* is a pattern[4] that specifies complex dependencies among values of attributes belonging to one or more real-world entities. Our reasoning for a DR is based on the concepts of *rule* [1] and *functional dependency* [3]. A rule is a common formalism for representing knowledge discovered through the application of data mining methods, like frequent itemset or association rule algorithms. A functional dependency is considered in the design of a relational DB schema in order to avoid redundancy and update anomalies. Both of them allow the definition of a set of attributes whose values determine the values of another set of attributes, and are usually specified as follows: $\{att_i, ..., att_k\} \Rightarrow \{att_m, ..., att_p\}$.

A DR also allows the specification of *determinant* and *determined* attribute sets, complex predicates involving these attribute sets, and the real-world entity type on which the DR holds (*target entity type*). Our contribution with the DR modeling on MASTER is to represent discovered patterns for the main real-world entities based mainly on the analysis of the aspects that surround them, as the aspects represent the relevant features of the trajectories, including spatial and temporal information. Suppose we had discovered a pattern in a MATs dataset stating that retired people in a small city usually move on foot when it is not raining. This pattern involves three aspects (*occupation*, *transportation* and *weather*) and could be specified by the following DR:

$DR_x$: `MAT | owner.is-a[description ='Person'] AND MAT_Aspect [description ='retired'].is-a[description ='occupation'] AND MAT_Aspect[NOT(description ='rainy')].is-a[description ='weather']` $\Rightarrow$ `MAT_Aspect[description = 'foot'].is-a[description = 'transportation']`

It shows that the DR is a pattern for a `MAT` (the target entity type). It also holds a pre-condition (before the implication) and a discovered data behaviour based on this pre-condition (after the implication).

We define a simple notation for a DR. It represents the DR three components (*target entity type*, *determinant* and *determined*), being similar to an association rule and a functional dependency. Therefore, this formalism tends to be easy to understand. In fact, we could adopted a rule specification language, like SWRL [10] and RIF [6]. However, they are verbose languages and would generate complex rule definitions. We now formally define a DR as well as its components.

**Definition 1.** ***(Filter)***. *A filter is a data restriction with the form att operator operand, where att is a required entity type attribute name that may be followed by an operator and an operand, with operator* $\in$ {=, $\neq$, >, >=, <, <=, `IS NULL`, `IS NOT NULL`}, *and operand is a required constant value, or another entity type attribute name if operator* $\notin$ {`IS NULL`, `IS NOT NULL`}.

[4] By pattern we mean an implicit (or hidden) regularity in the data.

**Definition 2.** ***(Predicate)****. A predicate is a boolean expression with the form of a path expression $pr_k = e_1.e_2...e_{n-1}.e_n$, $n > 0$, where each element $e_i \in pr_k$ is a MASTER relationship type that may be restricted by a filter optionally enclosed by a* `NOT` *logical operator and defined between brackets ('[', ']'), and $e_1$ is a relationship type connected to the MASTER target entity type.*

When a predicate has a filter with only the *att* part, we have an *existence* constraint w.r.t. the attribute. For example, the predicate `MO_Aspect[endTime]` states that the relationship *MO_Aspect* must hold the *endTime* attribute.

**Definition 3.** ***(Condition)****. A condition is a boolean expression with the form $pr_1$* `AND/OR` *$pr_2$* `AND/OR` *...* `AND/OR` *$pr_m$, $m > 0$, i.e., a non-empty set of predicates $\{pr_1, pr_2, ..., pr_m\}$ connected by the logical operators* `AND` *and* `OR`*.*

**Definition 4.** ***(Dependency Rule)****. A DR $dr_f$ is an expression with the form $te_x$* `|` *$c_i$* `=>` *$c_j$, where $te_x$ is a MASTER target entity type, with $te_x \in$ {*`MO`*,* `MAT`*,* `POINT`*,* `MOR`*}, $c_i$ is the pre-condition (or determinant), and $c_j$ is the condition regarding the discovered data regularity (or determined), and a $dr_f$ specification means that if $c_i$ is* `TRUE` *for a $te_x$ instance $i_x$, then $c_j$ is also* `TRUE` *for $i_x$.*

The aforementioned $DR_x$ is an example of DR specified according to Definition 4. We focus on semantic behaviours discovered for the aspects related to the main entity types (*target entity types*) of the MASTER model: moving objects (`MO`), multiple aspects trajectory (`MAT`), trajectory point (`POINT`), and moving object relationship (`MOR`). Because of this, a predicate must start with a relationship type connected to one of these entities.

A DR may also hold an existence constraint w.r.t. relationship types if their predicates have no filters. One example is $DR_y$: `MO | source OR target => MO_Aspect`. It states that when a moving object participates in a moving object relationship, it must be related to an aspect.

## 5 MASTER DR

This paper regards DRs for MAT data. As several patterns can be found over MATs, they can be valuable or not depending on their accuracy and temporal lifetime. Thus, we associate a *confidence* [1] and a *validity time* to each DR.

Figure 1 shows our MASTER extension to provide the representation of DRs and related concepts (the blue entities): the *MASTER DR*. The DR entity includes the aforementioned attributes. In the following, we define the DR entity.

**Definition 5.** ***(DR Entity)****. A DR Entity dre = (desc, startTime, endTime, confidence, EXT, PRE, POST, DS) is a discovered data pattern in a set of MAT datasets DS, with a description desc, a confidence and a validity time (startTime and endTime), as well as sets of predicates $PRE = \{pr_1, ..., pr_n\}$ and $POST = \{pr_1, ..., pr_m\}$ that specifies, respectively, its determinant and determined parts, and the set of sets $EXT = \{MOR \mid MO \mid MAT \mid POINT\}$, which are the sets of occurrences of MASTER entities on which the DR holds, being $MOR = \{mor_1$,*

*..., $mor_i$} the set of moving object relationships, MO = {$mo_1$, ..., $mor_j$} the set of moving objects, and so on.*

The *dre.EXT* sets are represented in MASTER DR as specialized entities (see Figure 1) that hold specific relationships with original MASTER entities $te_j$ that are the target of the DR. They are modeled as *many-to-many* relationships because the DR may be valid for several $te_j$ occurrences. The DR entity specializations are not exclusive as a same DR may serve, for example, as a pattern for a whole MAT in one context and a pattern for some MAT points in another context. A same pattern related to *weather*, for example, may occur during all the trajectory long or only for some of its points.

A DR may raise in several MAT datasets. We define a dataset as follows.

**Definition 6.** ***(Dataset Entity)****. A Dataset Entity dse = (desc, URL, DR) is a source of MAT data with a description desc, an URL with its location, and a set of discovered DR over it.*

As shown in Figure 1, a DR is composed of the *determinant* and *determined* conditions, which are sets of predicates. Thus, we also define a predicate entity.

**Definition 7.** ***(Predicate Entity)****. A Predicate Entity pe = (desc, owner, condition type, path, order, ASP, ASPT, parenthesisType, parenthesisAmount, logicalOperator) is part of a DR determinant or determined condition with a description desc, the DR the owns it (owner), the type of DR condition where it is inserted (determinant - 0; determined - 1) (condition type), the path expression that defines it, its order inside the condition, the optional sets of aspects (ASP) and aspect types (ASPT) occurrences on which the predicate holds, and optional attributes denoting whether the predicate is preceded or not by an open parenthesis (0) or succeeded by a close parenthesis (1) (parenthesis type), the amount of this parenthesis type (parenthesis amount), and whether the predicate is preceded by a logical operator: OR (0) or AND (1).*

For sake of understanding of the *Predicate Entity* definition suppose the following predicates that could be part of a determinant of a DR named $DR_z$:

`...((a.b.c[x = 1] OR d.e[y = 'q']) AND ...) ...`

This condition fragment has two predicates (`a.b.c[x = 1]` and `d.e[y = 'q']`), where the first one is preceded by two open parenthesis, and the second one is preceded by the `OR` logical operator and succeeded by one close parenthesis. In order to keep track of all the condition structure, we represent the first predicate as $pr_1$ = *('predicate$_1$', 'DR$_z$', 0, 'a.b.c[x = 1]', 1, NULL, {'aspectType$_i$'}, 0, 2, NULL)*, and the second predicate as $pr_2$ = *('predicate$_2$', 'DR$_z$', 0, 'd.e[y = 'q']', 1, {'aspect$_j$', 'aspect$_k$'}, {'aspectType$_l$'}, 1, 1, 0)*. We are also supposing that $pr_1$ is valid to an aspect type '*aspectType$_i$*', and $pr_2$ holds to the aspects '*aspect$_j$*' and '*aspect$_k$*', which belong to the aspect type '*aspectType$_l$*'.

This strategy to model predicates allows the representation of a DR condition composed of an arbitrary number of logical operators and parenthesis levels. We also associate a predicate with an aspect and/or an aspect type in order to allow queries like *"what MATs have patterns that enclose the aspect X?"* and *"what aspect types are more frequent in patterns for the moving object Y?"*.

## 6 Evaluation

We first present a case study with real trajectory data from the *SoBigData Consortium*, an European Union funding project related to Big Data social mining. We got a grant to access a repository with 8392 trajectories describing the movement of 129 users into Tuscany, Italy[5]. We analyzed two datasets of this repository as they provide more *aspects* associated to the trajectories:

– ***Diaries* dataset**: distance, average speed, day of the week, duration, day period and trajectory purpose (goal).
– ***MP* dataset**: goal, mean of transportation and duration.

The first dataset comprises people moving by car. In the second one, people had moved using different means of transportation. As both of them are stored into relational databases, we defined SQL grouping queries (queries with the `GROUP BY` $a_1, ..., a_n$ clause) to group data by sets of attributes' values $(a_1, ..., a_n)$ and verify whether a DR raises when they are viewed together (*e.g.*, to group trajectory data by *duration* and *goal*). We defined a group query for each incremental combination of the attributes' values that correspond to aspects, like *(goal, mean of transportation)*, $(goal, duration)$, ... , *(goal, mean of transportation, duration)* for the *MP* dataset. The result sets were analyzed manually.

Table 1 shows some discovered DRs on both datasets with high confidence, highlighting relevant patterns to be stored. All of these DRs can be represented by MASTER DR, demonstrating its usefulness in real data scenarios.

We also analyze query performance over MASTER DR when designed over open-source relational-based DB technologies: a *traditional DB management systems (DBMSs) (PostgreSQL)*, and a *NewSQL DB (VoltDB)*. We chose them because relational DBs is a dominant technology, and NewSQL is a emerging one. We run the experiments on a host of $Intel^R$ $Core^{TM}$ i5-8200 processor with 8 GB RAM (DDR4 1333Mhz), 240GB SSD and Xubuntu 20.04 Server LTS OS 64 bits. The generated relational schema for MASTER DR is shown in Figure 2.

We define five queries over the DBMSs covering the main MASTER DR entities and relationships, with different levels of complexity (from 3 to 10 table joins). We emulate 20 users randomly run each one of the queries over synthetic data in a DB with three sizes for each table: $10K$, $50K$ and $100K$ rows.

For the smallest DB size, PostgreSQL got a throughput of $3,613.88$ TPS (transactions per second), and VoltDB reached $6,146.41$ TPS. VoltDB has almost 100% more throughput than PostgreSQL for the smallest DB size. For the largest one, the difference falls to 14%. In short, the performance for querying the MASTER DR relational schema sounds good for both DBMSs, as they were able to consume hundreds of requests even for the largest DB size, with a light advantage of VoltDB. Despite the dataset sizes are not large, they were sufficient to evaluate the behavior of relational DBMSs in order to suggest appropriated DB technologies to support MASTER DR, as is the case of NewSQL DBs.

[5] *https://sobigdata.d4science.org/web/cityofcitizens/catalogue*

**Table 1.** Some discovered dependency rules.

| Dataset | Dependency Rule | Confidence |
|---|---|---|
| Diaries | goal = 'Supermarket' $\Rightarrow$ day period = '6-12' OR day period = '12-18' | 99 % |
| Diaries | goal = 'Work' OR goal = 'Service' OR goal = 'Study' $\Rightarrow$ NOT(day of the week = 'Sunday') | 87 % |
| Diaries | average speed > 80 AND (day of the week = 'Saturday' OR day of the week = 'Sunday') AND (day period = '0-6' OR day period = '18-24') $\Rightarrow$ goal = 'Leisure' | 72 % |
| MP | goal = 'Leisure' OR goal = 'Carburator Fixing' $\Rightarrow$ mean of transportation = 'Motorcycle' OR mean of transportation = 'Automobile' | 100 % |
| MP | goal = 'Pick up or drop out' $\Rightarrow$ mean of transportation = 'Automobile' | 100 % |
| MP | goal = 'Shopping' $\Rightarrow$ mean of transportation = 'Automobile' | 80 % |

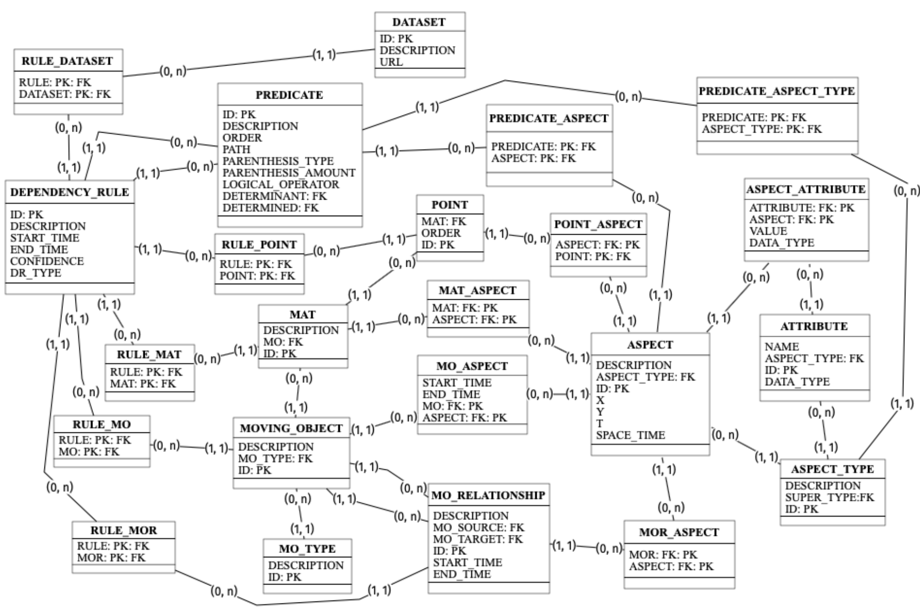


**Fig. 2.** A relational schema for MASTER DR

## 7 Conclusion

Proposals regarding trajectory pattern modeling are able to find out a limited set of (and simple) patterns as they consider only spatio-temporal attributes or limited semantic dimensions. This paper novels by proposing a conceptual model for DRs and a formalism to specify data patterns in the context of mobility data that are very rich in terms of semantics (MATs). We represent DRs as an extension of a conceptual model for MATs, thus enabling queries on trajectories and DRs in a joint way. This extended conceptual model combines simplicity by adding few entities to the original model, as well as the capability to represent patterns with different levels of complexity. A case study and a query performance analysis show that our conceptual model is useful and practicable. Future works include the analysis of other MAT datasets to better evaluate performance, expressiveness and limitations of MASTER DR. We also intend to simulate data insertions and updates over several DB technologies.

**Acknowledgments**. This work has been co-funded by the Brazilian agencies CNPQ and FAPESC (Project Match - Grant 2018TR 1266), and the European Union‘s Horizon 2020 research and innovation programme under GA N. 777695 (MASTER).

## References

1. Agrawal, R., et. al.: Mining Association Rules between Sets of Items in Large Databases. In: ACM SIGMOD Conference. pp. 207–216 (1993)
2. Alvares, L.O., et. al.: Dynamic Modeling of Trajectory Patterns using Data Mining and Reverse Engineering. In: ER Conference. pp. 149–154 (2007)
3. Bernstein, P.A., et. al.: A Unified Approach to Functional Dependencies and Relations. In: ACM SIGMOD Conference. pp. 237–245 (1975)
4. Bogorny, V., et. al.: A Conceptual Data Model for Trajectory Data Mining. In: GIS Conference. pp. 1–15 (2010)
5. Bogorny, V., et. al.: CONSTAnT - A Conceptual Data Model for Semantic Trajectories of Moving Objects. Transactions on GIS **18**(1), 66–88 (2014)
6. Damásio, C.V., et. al.: Declarative semantics for the rule interchange format production rule dialect. In: Semantic Web Conference. pp. 798–813 (2010)
7. Damiani, E., Frati, F.: Towards Conceptual Models for Machine Learning Computations. In: ER Conference. pp. 3–9 (2018)
8. Fileto, R., et. al.: Baquara: A Holistic Ontological Framework for Movement Analysis Using Linked Data. In: ER Conference. pp. 342–355 (2013)
9. Forlizzi, L., et. al.: A Data Model and Data Structures for Moving Objects Databases. In: ACM SIGMOD Conference. pp. 319–330 (2000)
10. Horrocks, I., Patel-Schneider, P.F.: A proposal for an owl rules language. In: WWW Conference. pp. 723–731 (2004)
11. Mello, R., et. al.: MASTER: A Multiple Aspect View on Trajectories. Trans. GIS **23**(4), 805–822 (2019)
12. Petry, L.M., et. al.: Towards semantic-aware multiple-aspect trajectory similarity measuring. Trans. GIS **23**(5), 960–975 (2019)
13. Rizzi, S.: UML-based Conceptual Modeling of Pattern-Bases. In: Workshop on Pattern Representation and Management - EDBT Conference (2004)
14. Ruback, L., et. al.: Enriching Mobility Data with Linked Open Data. In: IDEAS Symposium. pp. 173–182 (2016)
15. Spaccapietra, S., et. al.: A Conceptual View on Trajectories. Data Knowledge Engineering **65**(1), 126–146 (2008)
16. Zubcoff, J., et al.: Integrating Clustering Data Mining into the Multidimensional Modeling of Data Warehouses with UML Profiles. In: DaWaK Conference. pp. 199–208 (2007)